

NXP4330D/Q

Application Processor

Datasheet

Version 0.91

Release information

The following changes have been made to this document.

Release History

Date	Changes
7 June, 2013	V081 - First release V0.81
8 June, 2013	V082 - Added VDD Ball List Table - Added GND Ball List Table
13 June, 2013	V083 - Re-numbered all Figures - Re-numbered all Tables
26 June, 2013	V091 - Added Electrical Characteristics Section - Modified the MDIV value of System Control Section - Added Block Diagram of AC97 Controller

Proprietary Notice

Information in this document is provided solely to enable system and software implementers to use Nexell products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Nexell reserves the right to make changes without further notice to any products herein.

Nexell makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Nexell assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Nexell data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typical" must be validated for each customer application by customer's technical experts. Nexell does not convey any license under its patent rights nor the rights of others. Nexell products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Nexell product could create a situation where personal injury or death may occur. Should Buyer purchase or use Nexell products for any such unintended or unauthorized application, Buyer shall indemnify and hold Nexell and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Nexell was negligent regarding the design or manufacture of the part.

Copyright© 2013 Nexell Co.,Ltd. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Nexell.

Contact us

TEL: 82-31-698-7400

FAX: 82-31-698-7455

<http://www.nexell.co.kr>

Table of Contents

Section 1. Product Overview	18
1.1 Introduction.....	19
1.2 Key Features.....	20
1.3 Block Diagram.....	21
1.4 Brief Functional Specification	22
1.4.1 CPU	22
1.4.2 CLOCK & POWER MANAGEMENT	22
1.4.3 DMA.....	22
1.4.4 INTERRUPT CONTROLLER	22
1.4.5 TIMER & WATCHDOG TIMER	23
1.4.6 RTC	23
1.4.7 MEMORY CONTROLLER	23
1.4.8 GPIO CONTROLLER.....	23
1.4.9 ETHERNET MAC CONTROLLER.....	23
1.4.10 SD/MMC CONTROLLER.....	24
1.4.11 PPM.....	25
1.4.12 PWM.....	25
1.4.13 ADC.....	25
1.4.14 I2C.....	26
1.4.15 SPI/SSP	26
1.4.16 MPEG-TS.....	26
1.4.17 UART& ISO7816 SIM CARD INTERFACE	26
1.4.18 USB.....	27
1.4.19 I2S	28
1.4.20 AC97	28
1.4.21 SPDIF Tx, Rx.....	28
1.4.22 PDM.....	29
1.4.23 DISPLAY CONTROLLER	29
1.4.24 VIDEO POST PROCESSOR	30
1.4.25 VIDEO INPUT PROCESSOR.....	30
1.4.26 MULTI FORMAT MPEG CODEC	31
1.4.27 3D GRAPHIC CONTROLLER.....	32
1.4.28 SECURITY IP	33
1.4.29 UNIQUE CHIP ID	33
1.4.30 OPERATING CONDITIONS	33
1.4.31 PACKAGE	33
Section 2. I/O Pin Description.....	34
2.1 Mechanical Dimension.....	35

2.2	FCBGA Ball Map	37
2.3	I/O Function Description.....	42
2.3.1	BALL LIST TABLE	42
2.3.2	BALL LIST TABLE: SORTED BY FUNCTION	56
2.3.2.1	MCU-A.....	56
2.3.2.2	MCU-S.....	58
2.3.2.3	Digital RGB.....	60
2.3.2.4	HDMI	61
2.3.2.5	LVDS.....	61
2.3.2.6	MIPI DSI	61
2.3.2.7	MIPI CSI.....	62
2.3.2.8	VIP	62
2.3.2.9	Ethernet MAC	63
2.3.2.10	MPEG-TS Interface.....	64
2.3.2.11	UART_ISO7816	64
2.3.2.12	I2C.....	65
2.3.2.13	SPI/SSP.....	65
2.3.2.14	PWM	66
2.3.2.15	PPM	66
2.3.2.16	PDM	66
2.3.2.17	SPDIF.....	66
2.3.2.18	SD/MMC	66
2.3.2.19	USB2.0 HOST	67
2.3.2.20	USB2.0 HSIC HOST	68
2.3.2.21	USB2.0 OTG.....	68
2.3.2.22	I2S & AC97	68
2.3.2.23	ADC	68
2.3.2.24	ALIVE GPIO	69
2.3.2.25	JTAG	69
2.3.2.26	Crystal PLL & RTC	69
2.3.2.27	Miscellaneous.....	70
2.3.2.28	Power : VDD.....	70
2.3.2.29	Power : GND	71

Section 3. System Boot.....**72**

3.1	Overview	73
3.2	Functional Description	74
3.2.1	SYSTEM CONFIGURATION.....	74
3.3	External Static Memory Boot.....	76
3.3.1	EXTERNAL STATIC MEMORY BOOT FEATURES	76
3.3.2	EXTERNAL STATIC MEMORY BOOT SYSTEM CONFIGURATION.....	76
3.3.3	EXTERNAL STATIC MEMORY BOOT OPERATION.....	76

3.4 Internal ROM Boot.....	77
3.4.1 FEATURES	77
3.4.2 SYSTEM CONFIGURATION FOR THE INTERNAL ROM BOOTING	77
3.4.3 USB BOOT	78
3.4.3.1 Features.....	78
3.4.3.2 Operation	78
3.4.3.3 USB Descriptors.....	78
3.4.4 SDHCBOOT.....	79
3.4.4.1 Features.....	79
3.4.4.2 Operation	80
3.4.5 NANDBOOT WITH ERROR CORRECTION	81
3.4.5.1 Features.....	81
3.4.5.2 Operation	82
3.4.5.3 How to store User bootcode into the NAND flash memory.....	82
3.4.6 ADDITIONAL INFORMATION.....	83
3.4.6.1 ALIVE Power Control.....	83
3.4.6.2 Exception Vector Redirection	83
3.4.6.3 Parity generation for Error Correction.....	84
3.4.6.4 CRC32 Error Check.....	84
3.4.6.5 Data Decrytion with AES128 ECB mode to use hidden key.....	85
Section 4. System Control.....	86
4.1 Clock Manager.....	87
4.1.1 CLOCK MANAGER OVERVIEW.....	87
4.1.2 CLOCK MANAGER FEATURES	87
4.1.2.1 Clock Manager Block Diagram	87
4.1.3 CLOCK MANAGER FUNCTIONAL DESCRIPTION	88
4.1.3.1 PLL (Phase Locked Loop).....	88
4.1.3.2 Change PLL Value	93
4.1.3.3 Clock Generator	93
4.2 Power Manager	97
4.2.1 POWER MANAGER OVERVIEW	97
4.2.2 POWER DOWN MODE OPERATION.....	97
4.2.2.1 IDLE Mode	98
4.2.2.2 STOP Mode	99
4.2.2.3 SLEEP Mode 1, SLEEP Mode 2.....	99
4.2.2.4 GPIO as a Wake up Source	100
4.3 Reset Generation	102
4.3.1 POWER ON RESET SEQUENCE.....	102
4.3.2 SOFTWARE RESET AND GPIO RESET	102
4.3.3 WATCHDOG RESET.....	103
4.3.3.1 nPORST, Software Reset, Watchdog Reset and GPIO Reset	103

4.4 Register Summary.....	104
Section 5. Clock Generator	119
5.1 IP Clock Generator Overview.....	120
5.1.1 CLOCK GENERATOR LEVEL 0	120
5.1.1.1 Block Diagram.....	121
5.1.1.2 Register Summary.....	121
5.1.2 CLOCK GENERATOR LEVEL 1	123
5.1.2.1 Block Diagram.....	123
5.1.2.2 Register Summary.....	123
5.1.3 CLOCK GENERATOR LEVEL 2	127
5.1.3.1 Block Diagram.....	128
5.1.3.2 Register Summary.....	128
Section 6. System L2 Cache (PL-310 L2C)	135
6.1 Overview	136
6.1.1 FEATURES	136
6.1.2 BLOCK DIAGRAM	137
6.2 Functional Description	138
6.2.1 L2 CACHE USER CONFIG	138
6.2.2 INITIALIZATION SEQUENCE	138
6.3 Register Summary.....	139
Section 7. Secure JTAG	149
7.1 Overview	150
7.1.1 FEATURES	150
7.1.2 BLOCK DIAGRAM	150
7.1.3 SECURE JTAG USER CONFIG	150
Section 8. DMA	151
8.1 Overview	152
8.1.1 FEATURES	152
8.1.2 BLOCK DIAGRAM	153
8.2 Functional Description	154
8.2.1 SOFTWARE CONSIDERATIONS	154
8.2.2 PROGRAMMER'S MODEL	155
8.2.2.1 About the programmer's model.....	155
8.2.2.2 Programming the DMAC.....	156
8.2.2.3 Register Name Description.....	157
8.2.2.4 Peripheral DMA Request ID	160
8.2.2.5 Address generation	161

8.2.2.6	Scatter / gather	161
8.2.2.7	Interrupt requests.....	163
8.2.2.8	DMAC data flow.....	165
8.3	Register Summary.....	167
Section 9. Interrupt Controller.....	173	
9.1	Overview	174
9.1.1	FEATURES	174
9.1.2	BLOCK DIAGRAM	174
9.2	Programming sequence	176
9.2.1	INTERRUPT FLOW SEQUENCE USING AHB	176
9.2.2	FIQ INTERRUPT FLOW SEQUENCE.....	176
9.3	Interrupt Source	177
9.4	Register Summary.....	179
Section 10. Watch Dog Timer	181	
10.1	Overview	182
10.1.1	FEATURES	182
10.2	Functional Description	183
10.2.1	WATCHDOG TIMER OPERATION	183
10.2.2	WTDAT & WTCNT.....	183
10.2.3	CONSIDERATION OF DEBUGGING ENVIRONMENT	183
10.2.4	SPECIAL FUNCTION REGISTER.....	183
10.2.4.1	Memory map	183
10.2.4.2	Watchdog timer control(WTCON) register.....	184
10.2.4.3	Watchdog timer data(WTDAT) register.....	184
10.2.4.4	Watchdog timer count(WTCNT) register.....	184
10.2.4.5	Watchdog timer interrupt(WTCLRINT) register	184
10.3	Register Summary.....	185
Section 11. RTC	186	
11.1	Overview	187
11.1.1	FEATURES	187
11.1.2	BLOCK DIAGRAM	187
11.2	Functional Description	188
11.2.1	BACKUP BATTERY OPERATION.....	188
11.2.2	RTC OPERATION.....	188
11.2.3	ACCESSING THE RTC TIME COUNTER SETTING/READ REGISTER.....	188
11.2.4	INTERRUPT PENDING REGISTER	189
11.2.5	POWER MANAGER RESET TIME CONTROL	189
11.3	Register Summary.....	190

Section 12. Alive.....	192
12.1 Overview	193
12.1.1 FEATURES	193
12.2 Power Isolation.....	194
12.2.1 CORE POWER OFF.....	194
12.2.2 POWER GATING	194
12.3 Alive Registers.....	195
12.3.1 ALIVE GPIO DETECT REGISTERS	195
12.3.2 SCRATCH REGISTER	196
12.3.3 ALIVE GPIO CONTROL REGISTERS	196
12.4 Momentary Power Control.....	197
12.4.1 COREVDD POWERING ON	197
12.4.2 COREVDD POWERING OFF	197
12.5 SLEEP Mode	198
12.6 PMU (Power management Unit).....	199
12.6.1 OVERVIEW.....	199
12.6.2 POWER MODE TABLE	199
12.6.3 SWITCH CONTROL SEQUENCE.....	199
12.6.3.1 Power Off Sequence	199
12.6.3.2 Power On Sequence.....	199
12.7 Register Summary	200
Section 13. ID Register	223
13.1 Overview	224
13.1.1 FEATURES	224
13.2 Functional Description	225
13.2.1 AC TIMING	225
13.2.2 SENSE MODE TIMING.....	226
13.2.3 SCAN-LATCH MODE TIMING	227
13.2.4 STAND-BY MODE TIMING.....	227
13.2.5 PROGRAM MODE TIMING.....	228
13.3 Register Summary	229
13.4 Application Notes.....	233
Section 14. Memory Controller.....	234
14.1 Overview	235
14.1.1 UNIFIED MEMORY ARCHITECTURE (UMA)	235
14.1.2 BLOCK DIAGRAM	236
14.2 Functional Descriptions	237
14.2.1 MCU-A BANK FEATURE	237
14.2.2 MCU-S BANK FEATURE	237

14.2.3	MEMORY MAP.....	238
14.2.4	MCU-A ADDRESS MAPPING.....	239
14.2.4.1	Linear mapping.....	239
14.2.4.2	Interleaved mapping.....	239
14.2.5	LOW POWER OPERATION	240
14.2.5.1	AXI low power channel.....	240
14.2.5.2	Dynamic power down.....	240
14.2.5.3	Dynamic self refresh	240
14.2.5.4	Clock stop	240
14.2.5.5	Direct command	241
14.2.6	MCU-A APPLICATION NOTE	241
14.2.6.1	DREX initialization	241
14.2.6.2	DDRPHY initialization	245
14.2.7	DLL LOCK AND ZQ I/O CALIBRATION	251
14.2.7.1	Static Memory Map Shadow	251
14.2.7.2	Interface	253
14.2.8	NAND OVERVIEW.....	254
14.2.8.1	Normal Access Sequence	254
14.2.8.2	ECC (BCH).....	255
14.3	Register Summary.....	256
14.3.1	DREX REGISTER	256
14.3.2	DDRPHY REGISTER.....	269
14.3.3	MCU-S MEMORY CONTROL REGISTER	281
Section 15. GPIO Controller.....	291	
15.1	Overview	292
15.1.1	FEATURES	292
15.1.2	BLOCK DIAGRAM	293
15.2	Functional Description	294
15.2.1	INPUT OPERATION.....	294
15.2.2	OUTPUT OPERATION.....	294
15.2.3	ALTERNATE FUNCTION OPERATION.....	295
15.3	Register Summary.....	296
Section 16. Ethernet MAC.....	303	
16.1	Overview	304
16.1.1	MAC CORE FEATURES	304
16.1.2	DMA BLOCK FEATURES	304
16.1.3	BLOCK DIAGRAM	305
16.2	Register Summary	306
16.2.1	MAC DMA REGISTER DESCRIPTION.....	306
16.2.2	MAC CORE REGISTER DESCRIPTION.....	318

Section 17. SD/MMC Controller	339
17.1 Overview	340
17.1.1 FEATURES	340
17.1.1.1 Features of Mobile Storage Host	340
17.1.2 BLOCK DIAGRAM OF MOBILE STORAGE HOST	341
17.1.3 CLOCK PHASE SHIFTER	342
17.2 Register Summary	343
Section 18. Pulse Period Measurement (PPM)	358
18.1 Overview	359
18.1.1 FEATURES	359
18.1.2 BLOCK DIAGRAM	359
18.2 Functional Description	360
18.2.1 IR REMOTE PROTOCOL EXAMPLE	360
18.2.2 TIMING	361
18.2.3 FLOWCHART	361
18.3 Register Summary	362
Section 19. Pulse Width Modulation (PWM) Timer	363
19.1 Overview	364
19.1.1 FEATURES	365
19.1.2 BLOCK DIAGRAM	365
19.2 Functional Description	368
19.2.1 PRESCALER & DIVIDER	368
19.2.2 BASIC TIMER OPERATION	368
19.2.3 AUTO-RELOAD AND DOUBLE BUFFERING	369
19.2.4 INITIALIZE TIMER (SETTING MANUAL-UP DATA AND INVERTER)	370
19.2.5 OUTPUT LEVEL CONTROL	371
19.2.6 DEAD ZONE GENERATOR	371
19.2.7 TIMER INTERRUPT GENERATION	372
19.3 Register Summary	373
Section 20. Analog Digital Converter (ADC)	377
20.1 Overview	378
20.1.1 FEATURES	378
20.1.2 BLOCK DIAGRAM	378
20.2 Functional Description	379
20.2.1 I/O CHART	379
20.2.2 TIMING	380
20.2.3 ANALOG INPUT SELECTION TABLE	380
20.2.4 FLOWCHART	381

20.3 Register Summary.....	382
Section 21. I2C Controller	384
21.1 Overview	385
21.1.1 FEATURES	385
21.2 Functional Description	386
21.2.1 THE CONCEPT OF THE I2C-BUS	386
21.2.2 IC PROTOCOL.....	386
21.2.3 START/STOP OPERATION.....	387
21.2.4 DATA FORMAT.....	388
21.2.5 DATA TRANSFER	388
21.2.6 ARBITRATION	389
21.2.7 SYNCHRONIZATION	389
21.2.8 ACKNOWLEDGE	390
21.2.9 READ/WRITE OPERATION	390
21.2.10 CONFIGURATION I2C-Bus	391
21.2.11 OPERATIONS.....	392
21.2.12 INTERRUPT GENERATION	392
21.2.13 SYSTEM BUS RESET	392
21.3 Programming Guide.....	393
21.3.1 I2C-BUS INITIALIZE.....	393
21.3.2 COMMANDS	393
21.3.2.1 Master Transmitter Mode (M/Tx)	393
21.3.2.2 Master Receiver Mode (M/Rx)	394
21.3.2.3 Slave Transmitter Mode (S/Tx).....	395
21.3.2.4 Slave Receiver Mode (S/Rx)	395
21.4 Design Issues	397
21.4.1 SOFTWARE LAYER	397
21.4.2 DELAYS BETWEEN I2C-BUS COMMANDS	397
21.5 Register Summary.....	398
Section 22. SPI/SSP	400
22.1 Overview	401
22.1.1 FEATURES	401
22.1.2 BLOCK DIAGRAM	402
22.2 Functional Description	403
22.2.1 CLOCK GENERATION CONFIGURATION	403
22.2.1.1 Clock ratios.....	403
22.2.1.2 Programming the serial clock rate	403
22.2.2 OPERATION.....	404
22.2.2.1 Transmit and Receive operation	404
22.2.2.2 Reset operation	405

22.2.2.3	Interrupts	405
22.2.3	FRAME FORMAT	405
22.2.3.1	Texas Instruments synchronous serial frame format.....	406
22.2.3.2	Motorola SPI frame format	406
22.2.3.3	Motorola SPI Format with SPO=0, SPH=0	407
22.2.3.4	Motorola SPI Format with SPO=0, SPH=1	408
22.2.3.5	Motorola SPI Format with SPO=1, SPH=0	408
22.2.3.6	Motorola SPI Format with SPO=1, SPH=1	409
22.2.3.7	National Semiconductor Microwire frame format.....	410
22.3	Register Summary	412
Section 23. MPEG-TS Interface		415
23.1	Overview	416
23.1.1	FEATURES	416
23.2	Functional Description	417
23.2.1	TIMING	417
23.3	Register Summary	418
Section 24. UART_ISO7816		423
24.1	Overview	424
24.1.1	FEATURES	425
24.1.2	UART BLOCK DIAGRAM	427
24.1.3	IrDA SIR ENDEC FUNCTIONAL DESCRIPTION	429
24.1.3.1	IrDA SIR transmit encoder	429
24.1.3.2	IrDA SIR receive decoder.....	429
24.2	Operation.....	431
24.2.1	INTERFACE RESET.....	431
24.2.2	CLOCK SIGNALS.....	431
24.2.3	UART OPERATION	431
24.2.4	UART CHARACTER FRAME	434
24.2.5	IrDA DATA MODULATION.....	434
24.2.6	UART MODEM OPERATION	435
24.2.7	UART DMA INTERFACE	435
24.2.8	INTERRUPTS.....	437
24.2.9	ISO-7816.....	439
24.3	Register Summary	440
Section 25. USB2.0 OTG		449
25.1	Overview	450
25.1.1	FEATURES	450
25.1.2	BLOCK DIAGRAM	450

25.1.3 I/O PIN DESCRIPTION	451
25.2 Functional Description	452
25.2.1 END POINT PACKET SIZE.....	452
25.2.2 USB 5V POWER DETECTION	452
25.2.3 FORCE POWER DOWN.....	452
25.2.4 EXTERNAL CHARGE PUMP.....	453
25.2.5 MODES OF OPERATION.....	453
25.3 Programming User Config of PHY and OTG LINK.....	455
25.4 Register Summary.....	456
25.4.1 USB 2.0 OTG REGISTER MAP GROUPS.....	456
25.4.2 USB 2.0 OTG REGISTER MAP SUMMARY	457
25.4.3 CORE GLOBAL REGISTERS (USB_OTG_GCSR).....	463
25.4.4 HOST MODE REGISERS (USB_OTG_HMCSR).....	475
25.4.5 DEVICE MODE REGISERS (USB_OTG_DMCSR).....	482
25.4.6 POWER AND CLOCK GATING REGISERS (USB_OTG_PCGCCTL)	501
Section 26. USB2.0 HOST	502
26.1 Overview	503
26.1.1 FEATURES	503
26.1.2 BLOCK DIAGRAM	503
26.2 Functional Description	504
26.2.1 PROGRAMMING USER CONFIG OF PHY AND LINK IN EHCI OR OHCI	504
26.2.2 PROGRAMMING USER CONFIG OF PHY AND LINK IN HSIC	504
26.2.3 ATTENTION POINT OF HSIC PROGRAMMING	505
26.3 Register Summary	506
Section 27. I2S	509
27.1 Overview	510
27.1.1 FEATURES	510
27.2 Functional Description	511
27.2.1 BASIC CLOCK TREE	512
27.2.2 I2S-BUS FORMAT.....	512
27.2.3 SAMPLING FREQUENCY AND MASTER CLOCK.....	514
27.2.4 I2S CLOCK MAPPING TABLE	514
27.3 Programming Guide.....	515
27.3.1 Tx CHANNEL	515
27.3.2 Rx CHANNEL.....	517
27.3.3 I2S-CONTROLLER INITIALIZE.....	519
27.3.4 NOTES	519
27.4 Register Summary	521

Section 28. AC97.....	524
28.1 Overview	525
28.1.1 FEATURES	525
28.1.2 BLOCK DIAGRAM	525
28.2 Functional Description	526
28.2.1 INTERNAL DATA PATH	526
28.2.2 OPERATION FLOW CHART.....	526
28.2.3 AC-LINK DIGITAL INTERFACE PROTOCOL.....	527
28.2.3.1 AC-link Output Frame (SDATA_OUT)	528
28.2.3.2 AC-link Input Frame (SDATA_IN).....	529
28.2.4 AC97 POWER-DOWN	531
28.2.4.1 Powering Down the AC-Link	531
28.2.4.2 Waking Up the AC-Link - Wake Up Triggered by the AC97 Controller.....	531
28.2.4.3 Cold AC97 Reset.....	532
28.2.4.4 Warm AC97 Reset.....	532
28.2.4.5 AC97 State Diagram.....	533
28.3 Register Summary.....	534
Section 29. SPDIF TX.....	538
29.1 Overview	539
29.1.1 FEATURES	539
29.1.2 BLOCK DIAGRAM	539
29.2 Functional Description	541
29.2.1 DATA FORMAT OF SPDIF.....	541
29.2.1.1 Frame Format	541
29.2.1.2 Sub-frame Format (IEC 60958)	542
29.2.2 CHANNEL CODING	542
29.2.3 PREAMBLE	543
29.2.4 NON-LINEAR PCM ENCODED SOURCE (IEC 61937)	543
29.2.5 SPDIF OPERATION	544
29.2.6 SHADOWED REGISTER.....	545
29.3 Register Summary	546
Section 30. SPDIF RX.....	551
30.1 Overview	552
30.1.1 FEATURES	552
30.1.2 BLOCK DIAGRAM	552
30.2 Functional Description	553
30.3 Register Summary	554
Section 31. PDM.....	558

31.1	Overview	559
31.1.1	FEATURES	559
31.1.2	BLOCK DIAGRAM	559
31.1.3	PDM APPLICATION NOTE	559
31.1.3.1	Butterworth Filter Configuration	559
31.2	Register Summary	561
Section 32. Display Architecture.....		563
32.1	Overview	564
32.1.1	FEATURES	564
32.1.2	BLOCK DIAGRAM	564
32.2	TFT/MPU Interface	565
32.3	Register Summary	566
Section 33. Multi Layer Controller (MLC)		568
33.1	Overview	569
33.1.1	FEATURES	569
33.1.2	BLOCK DIAGRAM	570
33.2	Dual Register Set Architecture	571
33.3	MLC Global parameters.....	572
33.3.1	SCREEN SIZE.....	572
33.3.2	PRIORITY	572
33.3.3	FIELD MODE	573
33.3.4	BACKGROUND COLOR	573
33.4	Per-layer parameters.....	574
33.4.1	ENABLE	575
33.4.2	LOCK CONTROL	575
33.4.3	POSITION	575
33.4.4	PIXEL FORMAT	576
33.4.4.1	RGB layer format	576
33.4.4.2	Video layer format	577
33.4.4.3	Layer blending	578
33.4.5	ADDRESS GENERATION	580
33.4.5.1	RGB layer address generation	580
33.4.5.2	Video layer address generation	580
33.4.6	VIDEO LAYER SPECIFIC PARAMETERS	581
33.4.7	SCALE FUNCTION	581
33.4.8	COLOR CONTROL	582
33.4.8.1	Luminance Enhancement	582
33.4.8.2	Chrominance Enhancement	583
33.4.9	GAMMA CORRECTION	585
33.5	Clock Generation	587

33.6 Register Sumarry.....	588
Section 34. Display Controller (DPC)	599
34.1 Overview	600
34.1.1 FEATURES	600
34.1.2 BLOCK DIAGRAM	600
34.2 Sync Generator	601
34.2.1 CLOCK GENERATION.....	602
34.2.2 FORMAT	604
34.2.3 SYNC SIGNALS.....	606
34.2.4 SCAN MODE	611
34.2.5 DELAY.....	611
34.2.6 INTERRUPT	611
34.2.7 MPU (i80) TYPE SYNC SIGNALS	612
34.2.8 ODD/EVEN FIELD FLAG	613
34.3 Register Summary	614
Section 35. Scaler.....	624
35.1 Overview	625
35.1.1 FEATURES	625
35.1.2 BLOCK DIAGRAM	625
35.2 Functional Description	626
35.2.1 DIGITAL FILTER CHARACTERISTICS.....	626
35.2.1.1 Horizontal Filter (5-Tab FIR Filter) Frequency Response and Group Delay.....	626
35.2.1.2 Vertical Filter (3-Tab FIR Filter) Frequency Response and Group Delay.....	626
35.3 Programming Guide.....	628
35.3.1 CONFIGURATION.....	628
35.3.2 RUN.....	628
35.4 Register Summary	630
Section 36. LVDS.....	637
36.1 Overview	638
36.1.1 FEATURES	638
36.1.2 BLOCK DIAGRAM	638
36.2 Functional Description	639
36.2.1 LVDS DATA PACKING FORMAT	639
36.2.2 LVDS APPLICATION NOTE	640
36.2.3 SKEW CONTROL BETWEEN OUTPUT DATA AND CLOCK.....	640
36.2.4 ELECTRICAL CHARACTERISTICS.....	641
36.3 Register Summary	642
36.4 DisplayTop Register Summary	648

Section 37. HDMI.....	649
37.1 Overview	650
37.1.1 FEATURES	650
37.1.2 BLOCK DIAGRAM	651
37.2 Functional Description	652
37.2.1 SELECT RGB VIDEO DATA FOR HDMI.....	652
37.2.2 HDMI CONVERTER.....	652
37.2.3 HDMI LINK	653
37.2.3.1 Video Input Interface.....	653
37.2.3.2 Audio Input Interface	659
37.2.3.3 HPD	659
37.2.3.4 CEC	660
37.2.3.5 Interrupt Timing	660
37.2.3.6 HDCP KEY Management.....	660
37.3 Register Summary	662
37.3.1 CONTROL REGISTERS	662
37.3.2 CORE REGISTERS.....	666
37.3.3 AES REGISTERS.....	707
37.3.4 SPDIF REGISTERS.....	708
37.3.5 I2S REGISTERS	720
37.3.6 CEC REGISTERS	735
37.4 HDMI PHY	745
37.4.1 PHY CONFIGURATION CHANGE THROUGH APB	745
37.4.2 PHY READY SEQUENCE	746
37.4.3 HDMI PHY CONFIGURATION.....	746
37.4.4 HDMI PHY REGISTER SUMMARY.....	747
37.5 HDMI Application Sequences.....	749
37.6 DisplayTop Register Summary	750
Section 38. MIPI.....	751
38.1 Overview	752
38.1.1 FEATURES	752
38.2 DSIM 754	
38.2.1 BLOCK DIAGRAM OF MIPI DSI SYSTEM.....	754
38.2.2 INTERFACES AND PROTOCOL	756
38.2.3 CONFIGURATION.....	761
38.2.4 PLL	761
38.2.5 BUFFER	761
38.2.6 DSIM REGISTER SUMMARY.....	761
38.3 CSIS 778	
38.3.1 INTERFACES AND PROTOCOL	778
38.3.1.1 D-PHY layer FSM	778

38.3.1.2 PPI interface timing & protocol.....	778
38.3.2 CONFIGURATION.....	781
38.3.3 INTERRUPT	781
38.3.4 CLOCK SPECIFICATION.....	782
38.3.5 CSIS REGISTER SUMMARY.....	782
38.4 D-PHY	790
38.4.1 ARCHITECTURE	790
38.4.1.1 PLL and Clock Lane Connection	790
38.4.1.2 Data Lane Connection.....	790
38.4.1.3 IP Structure	791
38.4.1.4 Power Consumption.....	791
38.4.1.5 PAD Signals	792
38.4.1.6 Package and Board Connection Guideline.....	793
38.4.1.7 Core Interface Timing Diagram	794
Section 39. Video Input Processor (VIP).....	798
39.1 Overview	799
39.1.1 FEATURES	799
39.2 VIP Interconnection	800
39.2.1 BLOCK DIAGRAM	800
39.2.2 CLOCK GENERATION	800
39.3 Video Input Port	801
39.3.1 BLOCK DIAGRAM	801
39.3.2 SYNC GENERATION	801
39.3.3 EXTERNAL DATA VALID AND FIELD	805
39.3.4 DATA ORDER	806
39.3.5 STATUS	806
39.3.6 FIFO CONTROLS	807
39.3.7 RECOMMEND SETTING FOR VIDEO INPUT PORT	807
39.4 CLIPPER & DECIMATOR	808
39.4.1 CLIPPING & SCALE-DOWN	808
39.4.2 OUTPUT DATA FORMAT	809
39.4.3 FLIP & ROTATION	811
39.4.4 INTERLACE SCAN MODE.....	811
39.4.5 PIXELS ALIGNMENT.....	811
39.5 Interrupt Generation	812
39.6 Register Summary	813
Section 40. Multi-Format Video Codec.....	823
40.1 Overview	824
40.1.1 FEATURES	824
40.2 Functional Description	825

40.2.1 LIST OF VIDEO CODECS	825
40.2.2 SUPPORTED VIDEO ENCODING TOOLS	825
40.2.3 SUPPORTED VIDEO DECODING TOOLS	826
40.2.4 SUPPORTED JPEG TOOLS.....	828
40.2.5 NON-CODEC RELATED FEATURES	828
Section 41. 3D Graphic Engine.....	829
41.1 Overview	830
41.1.1 PIXEL PROCESSOR FEATURES	830
41.1.2 GEOMETRY PROCESSOR FEATURES	831
41.1.3 LEVEL 2 CACHE CONTROLLER FEATURES.....	831
41.1.4 MMU.....	831
41.1.5 PMU.....	831
41.2 Operation.....	832
41.2.1 CLOCK	832
41.2.2 RESET.....	832
41.2.3 INTERRUPT	832
Section 42. Crypto Engine.....	833
42.1 Overview	834
42.1.1 FEATURES	834
42.1.2 BLOCK DIAGRAM	834
42.2 Functional Description	836
42.3 Register Summary.....	837
Section 43. Electrical Characteristics.....	843
43.1 Absolute Maximum Ratings	844
43.2 Recommended Operating Conditions	845
43.3 D.C. Electrical Characteristics	847

Section 1. Product Overview

1.1 Introduction

NXP4330D/Q is a system-on-a-chip (SoC) based on the 32-bit RISC processor for tablets and cell-phones. Designed with the 28 nm low power process, features of NXP4330D/Q include:

- Cortex-A9 Dual/Quad core CPU
- Highest memory bandwidth
- Full HD display
- 1080p 60 frame video decoding and 1080p 30 frame encoding hardware
- 3D graphics hardware
- High-speed interfaces such as eMMC4.5 and USB 2.0

NXP4330D/Q uses the Cortex-A9 quad core, which is 50 % overall performance higher than Cortex-A8 core and its speed is 1.4GHz. It provides 6.4 GB/s memory bandwidth for heavy traffic operations such as 1080p video encoding and decoding, 3D graphics display and high resolution image signal processing with Full HD display. The application processor supports dynamic virtual address mapping, which helps software engineers to fully utilize the memory resources with ease.

NXP4330D/Q provides the best 3D graphics performance with wide range of APIs, such as OpenGL ES1.1, 2.0. Superior 3D performance fully supports Full HD display. The native dual display, in particular, supports Full HD resolution of a main LCD display and 1080p 60 frame HDTV display throughout HDMI, simultaneously. Separate post processing pipeline enables NXP4330D/Q to make a real display scenario.

NOTE)

- NXP4330D : Cortex-A9 Dual Core CPU / NXP4330Q : Cortex-A9 Quad Core CPU
- NXP4330D and NXP4330Q are pin to pin compatible

1.2 Key Features

- 28nm, HKMG (High-K Metal Gate) Process Technology
- 513 pin FCBGA Package, 0.65nm Ball Pitch, 17x17mm Body size
- Cortex-A9 Dual/Quad Core CPU @ 1.4Ghz
- High Performance 3D Graphic Accelerator
- Full-HD Multi Format Video Codec
- Supports various memory : x32 LPDDR2/3, DDR3 up to 800Mhz
- Supports MLC/SLC NAND Flash with Hardwired ECC algorithm (4/8/12/16/24/40/60bit)
- Supports Dual Display up to 2048x1280, TFT-LCD, LVDS, HDMI 1.4a, MIPI-DSI output
- Supports 2x ITUR.BT 656 Parallel Video Interface and MIPI-CSI
- Supports 10/100/1000M-bit Ethernet MAC
- Supports 3ch SD/MMC, 6ch UARTs, 32ch DMAs, 4ch Timer, Interrupt Controller, RTC
- Supports 3ch I2S, SPDIF Rx/Tx, 3ch I2C, 3ch SPI, 8ch 12bit ADC, 3ch PWM and GPIOs,
- Supports MPEG-TS Interface and MPEG-TS HW Parser
- Supports USB 2.0 Host, USB 2.0 OTG, USB HSIC
- Supports Security functions (AES, DES/TDES, SHA-1, MD5 and PRNG) and Secure JTAG
- Supports various Power Mode (Normal, Sleep, Deep-Sleep, Stop)
- Supports various boot modes including NAND (with ECC detection and correction), SPI Flash/EEPROM, NOR, USB and UART

1.3 Block Diagram

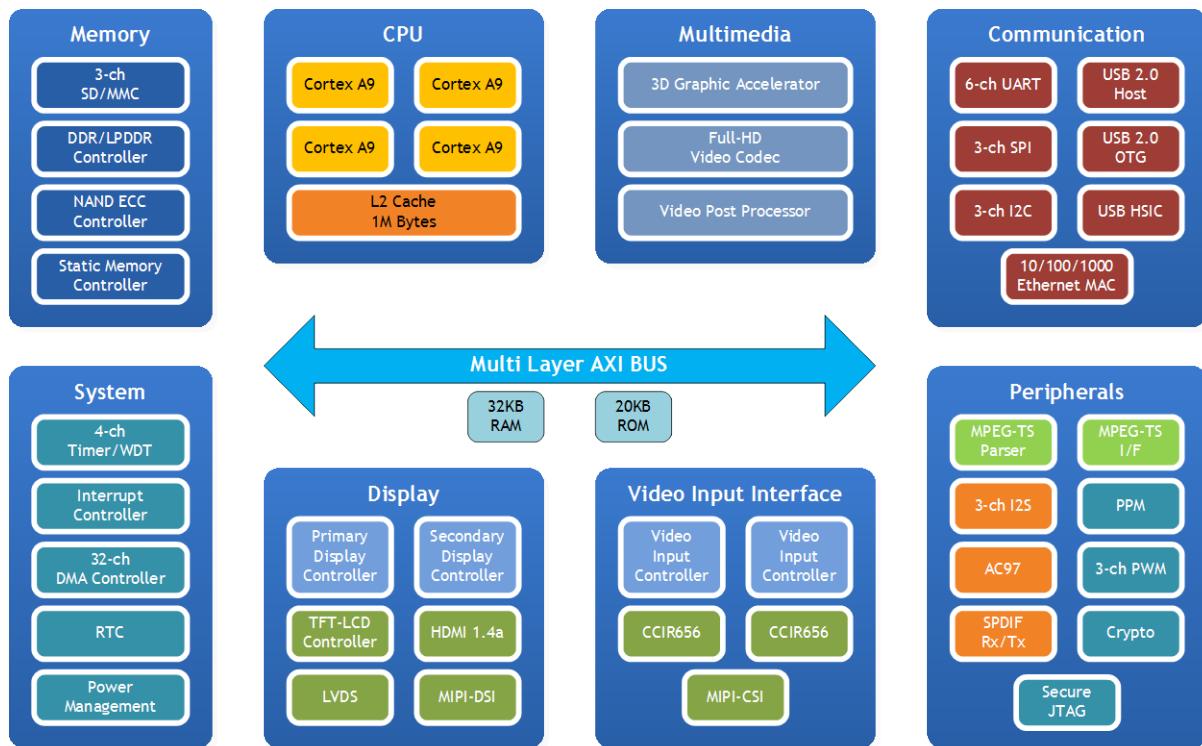


Figure 1-1. Block Diagram

1.4 Brief Functional Specification

1.4.1 CPU

- Cortex-A9 Dual/Quad Core @ 1.4GHz
- L1 Cache
 - 32Kbyte I-Cache, 32Kbyte D-Cache
- L2 Cache
 - 1Mbyte Shared Cache
- Co-Processor
 - VFP (Vector Floating Point Processor), Neon Processor

1.4.2 Clock & Power Management

- 4 Spread-Spectrum PLLs
- External Crystal : 24Mhz, 32.768Khz
- Supports for various power mode
 - Normal, Idle, Stop
 - Suspend to RAM (Sleep, Deep Sleep)

1.4.3 DMA

- 32-ch DMAs
- Operation Mode
 - Memory-to-Memory Transfer
 - Memory to IO Transfer, IO to Memory Transfer

1.4.4 Interrupt Controller

- Vectored Interrupt Controller
- Supports 64-ch Interrupt Sources
- Supports following features
 - fixed hardware interrupt priority levels
 - programmable interrupt priority levels
 - hardware interrupt priority level masking
 - programmable interrupt priority level masking
 - IRQ and FIQ generation
 - software interrupt generation
 - test registers
 - raw interrupt status
 - interrupt request status

1.4.5 Timer & Watchdog Timer

- 4-ch Timer with Watchdog Timer
- Normal interval timer mode with interrupt request
- Internal reset signal is activated when the timer count value reaches 0 (time-out)
- Level-triggered interrupt mechanism

1.4.6 RTC

- 32bit Counter
- Support Alarm Interrupt

1.4.7 Memory Controller

- System Memory Controller
 - Supports LPDDR2/LPDDR3/DDR3 SDRAM up to 2Gbytes
 - Supports 1.2V ~ 1.5V power
 - Max Operation Frequency : 800Mhz
 - Data Bus width : 32-bit
- Static Memory Controller
 - Multiplexed Address : up to 24-bit
 - SRAM, ROM and NAND Flash
 - Burst Read/Write
- NAND Flash Controller
 - Supports SLC/MLC NAND Flash
 - Supports MLC NAND Boot
 - Hardwired ECC Algorithm
 - 4/8/12/16/24/40/60-bit BCH Error Correction

1.4.8 GPIO Controller

- Various GPIO Interrupt Modes
 - Rising Edge, Falling Edge, High Level, Low Level Detection
- Individual Interrupt Generation

1.4.9 Ethernet MAC Controller

- Standard Compliance
 - IEEE 802.3az-2010, for Energy Efficient Ethernet (EEE)
 - RGMII specification version 2.6 from HP/Marvell
- MAC supports the following features
 - 10, 100, and 1000 Mbps data transfer rates with the following PHY interfaces:
 - RGMII interface to communicate with an external gigabit PHY
 - Full-duplex operation:

- IEEE 802.3x flow control automatic transmission of zero-quanta Pause frame on flow control input de-assertion
- Optional forwarding of received Pause frames to the user application
- Half-duplex operation:
 - CSMA/CD Protocol support
 - Flow control using backpressure support (based on implementation-specific white papers and UNH Ethernet Clause 4 MAC Test Suite - Annex D)
 - Frame bursting and frame extension in 1000 Mbps half-duplex operation
- Preamble and start of frame data (SFD) insertion in Transmit path
- Preamble and SFD deletion in the Receive path
- Automatic CRC and pad generation controllable on a per-frame basis
- Automatic Pad and CRC Stripping options for receive frames
- Flexible address filtering modes, such as:
 - Up to 31 additional 48-bit perfect (DA) address filters with masks for each byte
 - Up to 96 additional 48-bit perfect (DA) address filters that can be selected in blocks of 32 and 64 registers
 - Up to 31 48-bit SA address comparison check with masks for each byte
 - 64-bit, 128-bit, or 256-bit Hash filter (optional) for multicast and unicast (DA) addresses
 - Option to pass all multicast addressed frames
 - Promiscuous mode to pass all frames without any filtering for network monitoring
 - Pass all incoming packets (as per filter) with a status report
- Programmable frame length to support Standard or Jumbo Ethernet frames with up to 16 KB of size
- Programmable Interframe Gap (IFG) (40-96 bit times in steps of 8)
- Option to transmit frames with reduced preamble size
- Separate 32-bit status for transmit and receive packets
- IEEE 802.1Q VLAN tag detection for reception frames
- Additional frame filtering:
 - VLAN tag-based: Perfect match and Hash-based (optional) filtering
 - Layer 3 and Layer 4-based: TCP or UDP over IPv4 or IPv6
- Separate transmission, reception, and control interfaces to the application
- MDIO master interface (optional) for PHY device configuration and management
- Standard IEEE 802.3az-2010 for Energy Efficient Ethernet
- CRC replacement, Source Address field insertion or replacement, and VLAN insertion, replacement, and deletion in transmitted frames with per-frame control
- Programmable watchdog timeout limit in the receive path

1.4.10 SD/MMC Controller

- 3 Independent SD/MMC Controller and Ports
- Secure Digital Memory (SD mem- version 3.0)
- Secure Digital I/O (SDIO - version 3.0)
- Consumer Electronics Advanced Transport Architecture (CE-ATA - version 1.1)

- Multimedia Cards (MMC - version 4.41, eMMC 4.5)
- Supports following features of MMC4.41
- Support following features of eMMC4.5
- Support clock speed up to 50 MHz
- Support PIO and DMA mode data transfer
- Support 1/4/8-bit data bus widths
 - Overlay SPI signals to same GPIOs from SSP/SPI controller

1.4.11 PPM

- Pulse Period Measurement for IR remote receiver

1.4.12 PWM

- 3-ch PWM Controller
- Five 32-bit Timers
- Two 8-bit Clock Prescalers providing first level of division for the PCLK, Five Clock Dividers and Multiplexers providing second level of division for the Prescaler clock and Two External Clocks
- Programmable Clock Select Logic for individual PWM Channels
- Four Independent PWM Channels with Programmable Duty Control and Polarity
- Static Configuration: PWM is stopped
- Dynamic Configuration: PWM is running
- Supports Auto-Reload Mode and One-Shot Pulse Mode
- Supports for two external inputs to start PWM
- Dead Zone Generator on two PWM Outputs
- Supports DMA Transfers
- Optional Pulse or Level Interrupt Generation
- The PWM has two operation modes:
 - Auto-Reload Mode
 - Continuous PWM pulses are generated based on programmed duty cycle and polarity
 - One-Shot Pulse Mode
 - Only one PWM pulse is generated based on programmed duty cycle and polarity

1.4.13 ADC

- 8-ch analog input port
- Supports following features
 - Resolution: 12-bit
 - Conversion rate : 1MSPS
 - Power consumption
 - 1.0 mW (Fs = 1MSPS) @ Normal operation mode Typ.

- 0.005 mW @ Power down mode Typ.

- Input range: 0 ~ AVDD18
- Input frequency: up to 100kHz
- Digital output: CMOS Level (0 ~ AVDD10)

1.4.14 I2C

- 3-ch I2C bus controller
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; master can operate as master-transmitter or as master-receiver
- Serial, 8-bit oriented, bi-directional data transfers can be made at up to 100kbit/s in the Standard-mode, up to 400kbit/s in the Fast-mode
- The number of ICs that can be connected to the same bus is limited only by a maximum bus capacitance of 400pF
- Repeated START and early termination function are not support
- High speed mode, combined format, 10bit address are not supported

1.4.15 SPI/SSP

- 3-ch SPI Controller
- Master or slave operation.
- Programmable clock bit rate and prescale.
- Separate transmit and receive first-in, first-out memory buffers, 16 bits wide, 8 locations deep.
- Programmable choice of interface operation, SPI, Microwire, or TI synchronous serial.
- Programmable data frame size from 4 to 16 bits.
- Independent masking of transmit FIFO, receive FIFO, and receive overrun interrupts.
- Internal loopback test mode available.
- Support for Direct Memory Access (DMA).

1.4.16 MPEG-TS

- Supports Serial & Parallel MPEG-TS Interface
- Supports Hardwired MPEG2-TS parser for Set-top and IPTV

1.4.17 UART& ISO7816 Sim Card Interface

- 6-ch UART controller
- Programmable use of UART or IrDA SIR input/output.
- Separate 32×8 transmit and 32×12 receive First-In, First-Out (FIFO) memory buffers to reduce CPU interrupts.
- Programmable FIFO disabling for 1-byte depth.

- Programmable baud rate generator. This enables division of the reference clock by (1×16) to (65535×16) and generates an internal ×16 clock. The divisor can be a fractional number enabling you to use any clock with a frequency >3.6864MHz as the reference clock.
- Standard asynchronous communication bits (start, stop and parity). These are added prior to transmission and removed on reception.
- Independent masking of transmit FIFO, receive FIFO, receive timeout, modem status, and error condition interrupts.
- Support for Direct Memory Access (DMA).
- False start bit detection.
- Line break generation and detection.
- Support of the modem control functions CTS, DCD, DSR, RTS, DTR, and RI.
- Programmable hardware flow control.
- Fully-programmable serial interface characteristics:
 - data can be 5, 6, 7, or 8 bits
 - even, odd, stick, or no-parity bit generation and detection
 - 1 or 2 stop bit generation
 - baud rate generation, dc up to UARTCLK/16
- IrDA SIR ENDEC block providing:
 - programmable use of IrDA SIR or UART input/output
 - support of IrDA SIR ENDEC functions for data rates up to 115200 bps half-duplex
 - support of normal 3/16 and low-power (1.41-2.23μs) bit durations
 - programmable division of the UARTCLK reference clock to generate the appropriate bit duration for low-power IrDA mode.
- Identification registers that uniquely identify the UART. These can be used by an operating system to automatically configure itself.

1.4.18 USB

- 1-ch USB 2.0 Host and 1-ch USB2.0 HSIC Host
 - fully compliant with the Universal Serial Bus Specification, Revision 1.1, Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 2.0, and the openHCI: Open Host Controller Interface Specification for USB, Release 1.0a. The controller supports high-speed, 480-Mbps transfers (40 times faster than USB 1.1 full-speed mode) using an EHCI Host Controller, as well as full and low speeds through one or more integrated OHCI Host Controllers.
 - At the USB 2.0 physical interface, the controller provides the following:
 - UTMI: UTMI+ Level 3, Revision 1.0
 - High-Speed Inter-Chip (HSIC), Version 1.0
 - Supports ping and split transactions
 - UTMI/UTMI+ PHY interface clock supports 30-MHz operation for a 16-bit interface or 60-MHz operation for an 8-bit interface
 - Heterogeneous selection of UTMI+ or HSIC interfaces per port using strap pins. In Heterogeneous mode, only the 8-bit interface (60 MHz) is supported.
- 1-ch USB 2.0 OTG Controller

- supports both device and host functions and complies fully with the On-The-Go Supplement to the USB 2.0 Specification, Revision 1.3a and Revision 2.0. It can also be configured as a host-only or device-only controller, fully compliant with the USB 2.0 Specification.
- Complies with the On-The-Go Supplement to the USB 2.0 Specification (Revision 1.3)
- Complies with the On-The-Go Supplement to the USB 2.0 Specification (Revision 2.0)
- Software configurable to OTG1.3 and OTG2.0 modes of operation
- Support for the following speeds:
 - High-Speed (HS, 480-Mbps),
 - Full-Speed (FS, 12-Mbps) and
 - Low-Speed (LS, 1.5-Mbps) modes
- Multiple options available for low power operations
- Multiple DMA/non DMA mode access support on the application side
- Multiple Interface support on the MAC-Phy
- Supports 16 bidirectional endpoints, including control endpoint 0.
- Supports Session Request Protocol (SRP)
- Supports Host Negotiation Protocol (HNP)
- Supports up to 16 host channels. In Host mode, when the number of device endpoints to be supported is more than the number of host channels, software can reprogram the channels to support up to 127 devices, each having 32 endpoints (IN + OUT), for a maximum of 4,064 endpoints.
- Includes automatic ping capabilities

1.4.19 I2S

- 3-ch I2S Controller for 5.1ch Audio output
- 16bit/24bit Master & Slave Mode
- Supports various interface mode
 - I2S, Left-justified, Right-Justified, DSP mode
- Supports TDM mode for Digital MIC interface
- Supports SPDIF Rx/Tx

1.4.20 AC97

- 1-Ch AC97
- Independent channels for stereo PCM In, stereo PCM Out, mono MIC In
- DMA-based operation and interrupt based operation
- All of the channels support only 16-bit samples
- Variable sampling rate AC97 Codec interface (48 kHz and below)
- 16-bit, 16 entry FIFOs per channel
- Only primary Codec support

1.4.21 SPDIF Tx, Rx

- SPDIF Tx

- Supports linear PCM up to 24-bit per sample
- Supports Non-linear PCM formats such as AC3, MPEG1 and MPEG2
- 2x24-bit buffers which is alternately filled with data
- SPDIF Rx
 - Serial, unidirectional, self-clocking interface
 - Single wire-single signal interface
 - Easy to work because it is polarity independent

1.4.22 PDM

- Supports receiving 2 channel audio data with 1 data pin
- 1 output clock pin
- 2 data pins (total 4 channel accepts available)
- Fixed output clock frequency
- Supports select of the timing of data capture
- Supports DMA interface
- Supports a user configuration of Coefficient. (Butterworth Low-pass Filter)

1.4.23 Display Controller

- Supports Dual Display
- Supports 3 Layers, Gamma Correction and Color Control (Brightness, Contrast, Hue and Saturation)
- Supports various Pixel Format
 - RGB/BGR 444,555,565,888 with/without Alpha channel
- Resolution
 - Up to 2048x1280 @60hz
- Supports various LCD
 - I80 Interface, RGB, Serial RGB, LVDS output
 - Supports MIPI-DSI 4 data lanes
- HDMI Interface
 - HDMI 1.4a, HDCP 1.4 Complaint
 - Supports Video format:
 - 480p @59.94Hz/60Hz, 576p@50Hz
 - 720p @50Hz/59.94Hz/60Hz
 - 1080i @50Hz/59.94Hz/60Hz
 - 1080p @50Hz/59.94Hz/60Hz
 - Primary 3D Video Formats
 - Other various formats up to 148 MHz Pixel Clock
 - Supports Color Format : 4:4:4 RGB/YCbCr , 4:2:2 YCbCr
 - Pixel Repetition : Up to x4
 - Supports Bit Per Color : 8bit, 10bit ,12bit (Note: 16bit not supported)
 - Dedicated block for CEC function

- Supports : Linear-PCM, Non-linear PCM and high-bitrate audio formats (Audio Sample packets and HBR packets for audio transmission)
- Integrated HDCP Encryption Engine for Video/Audio content protection (Authentication procedure are controlled by S/W, not by H/W)
- A dedicated CEC module (Separated for power/clock domain separation)
- SPDIF Interface and I2S interface for Audio Input
- Supports level-triggered Interrupt and SFR for HPD
- Supports AES KEY Decryption Function for external HDCP Key management
- LVDS Interface
 - Output clock range: 30M to 90MHz
 - 35:7 data channel compression up to 630Mbps on each LVDS channel
 - Power down mode
 - Up to 393.75Mbytes/sec bandwidth
 - Falling clock edge data strobe
 - Narrow bus reduces cable size and cost
 - PLL requires no external component
 - 6 LVDS output channels (5 data channels, 1 clock channel)
- MIPI-DSI
 - Complies to MIPI DSI Standard Specification V1.01r11
 - Maximum resolution ranges up to WUXGA (1920x1200)
 - Supports 1, 2, 3, or 4 data lanes
 - Supports pixel format: 16bpp, 18bpp packed, 18bpp loosely packed (3 byte format), and 24bpp
 - Interfaces
 - Complies with Protocol-to-PHY Interface (PPI) in 1.5Gbps MIPI D-PHY
 - Supports RGB Interface for Video Image Input from display controller
 - Supports I80 Interface for Command Mode Image input from display controller
 - Supports PMS control interface for PLL to configure byte clock frequency
 - Supports Prescaler to generate escape clock from byte clock

1.4.24 Video Post Processor

- 3D De-interlace Controller
- Fine Scalar for video : Poly-phase filter

1.4.25 Video Input Processor

- Max. 8192x8192 resolution support
- Supports x2 8bit BT656, 601 format
- Supports MIPI-CSI
 - General Features
 - Support primary and secondary Image format
 - YUV420, YUV420(Legacy), YUV420(CSPS), YUV422 of 8-bits and 10-bits

- RGB565, RGB666, RGB888
- RAW6, RAW7, RAW8, RAW10, RAW12, RAW14
- Compressed format : 10-6-10, 10-7-10, 10-8-10
- All of User defined Byte-based Data packet
- Support embedded byte-based non Image data packet and generic short packets.
- Compatible to PPI(Protocol-to-PHY Interface) in MIPI D-PHY Specification
- Support 4 channel virtual channel or data interleave
- Standard Compliance
 - Compliant to MIPI CSI2 Standard Specification V1.01r06
 - D-phy standard specification V1.0

1.4.26 Multi Format MPEG codec

- Decoder
 - H.264
 - BP, MP, HP profile, Level 4.2 up to 1920x1080, 50Mbps
 - MPEG4 ASP
 - Divx, Xvid Advanced Simple Profile up to 1920x1080, 40Mbps
 - H.263
 - Profile3 up to 1920x1080, 20Mbps
 - VC-1
 - SP/MP/AP profile, Level 3 up to 1920x1080, 2048x1024, 45Mbps
 - MPEG-1/2
 - Main Profile, High Level up to 1920x1080, 80Mbps
 - VP8
 - up to 1920x1080, 20Mbps
 - Theora
 - up to 1280x720, 20Mbps
 - AVS
 - Jizhun Profile, Level 6.2 up to 1920x1080, 40Mbps
 - RV8/9/10
 - up to 1920x1080, 40Mbps
 - MJPEG
 - Baseline profile up to 8192x8192
- Encoder
 - H.264
 - Baseline Profile, Level 4.0 up to 1080p, 20Mbps
 - MPEG4
 - Simple Profile, Level 5.6 up to 1080p, 20Mbps
 - H.263

- Profile3, Level 70 up to 1080p, 20Mbps
- MJPEG
- Baseline Profile up to 8192x8192

1.4.27 3D Graphic Controller

- Supports OpenGL|ES 1.0 and 2.0
- Supports OpenVG 1.1
- GPU is a hardware accelerator for 2D and 3D graphics systems.
- The GPU consists of:
 - one to four Pixel Processors (PPs)
 - a Geometry Processor (GP)
 - a Level 2 Cache Controller (L2)
 - a Memory Management Unit (MMU) for each GP and PP included in the GPU
 - a Power Management Unit (PMU).
- Pixel processor features
 - each pixel processor used processes a different tile, enabling a faster turnaround
 - programmable fragment shader
 - alpha blending
 - complete non-power-of-2 texture support
 - cube mapping
 - fast dynamic branching
 - fast trigonometric functions, including arctangent
 - full floating-point arithmetic
 - framebuffer blend with destination Alpha
 - indexable texture samplers
 - line, quad, triangle and point sprites
 - no limit on program length
 - perspective correct texturing
 - point sampling, bilinear, and trilinear filtering
 - programmable mipmap level-of-detail biasing and replacement
 - stencil buffering, 8-bit
 - two-sided stencil
 - unlimited dependent texture reads
 - 4-level hierarchical Z and stencil operations
 - Up to 512 times Full Scene Anti-Aliasing (FSAA). 4x multisampling times 128xsupersampling
 - 4-bit per texel compressed texture format.
- Geometry processor features
 - programmable vertex shader
 - flexible input and output formats
 - autonomous operation tile list generation
 - indexed and non-indexed geometry input
 - primitive constructions with points, lines, triangles and quads.

- Level 2 cache controller features
 - sizes of 32KB
 - 4-way set-associative
 - supports up to 32 outstanding AXI transactions
 - implements a standard pseudo-LRU algorithm
 - cache line and line fill burst size is 64 bytes
 - supports eight to 64bytes uncached read bursts and write bursts
 - 128-bit interface to memory sub-system
 - support for hit-under-miss and miss-under-miss with the only limitation of AXI ordering rules.
- MMU features
 - accesses control registers through the bus infrastructure to configure the memory system.
 - each processor has its own MMU to control and translate memory accesses that the GPU initiates.
- PMU features
 - programmable power management
 - powers up and down each GP, PP and Level 2 cache controller separately
 - controls the clock, isolation and power of each device
 - provides an interrupt when all requested devices are powered up

1.4.28 Security IP

- On-chip secure boot ROM/RAM
- Hardware Crypto Accelerator
 - DES/TDES, AES, SHA-1, MD5 and PRNG
- Supports Secure JTAG

1.4.29 Unique Chip ID

- Supports 128-bit Unique Chip ID register

1.4.30 Operating Conditions

- Operation Voltage
 - Core : 1.0V
 - CPU : 1.0V ~ 1.3V
 - DDR Memory : 1.2~1.5V
 - I/O : 3.3V
- Operation Temperature
 - T.B.D

1.4.31 Package

- 513 pin FCBGA
- Ball Pitch: 0.65 nm
- Body Size : 17x17 mm

Section 2. I/O Pin Description

2.1 Mechanical Dimension

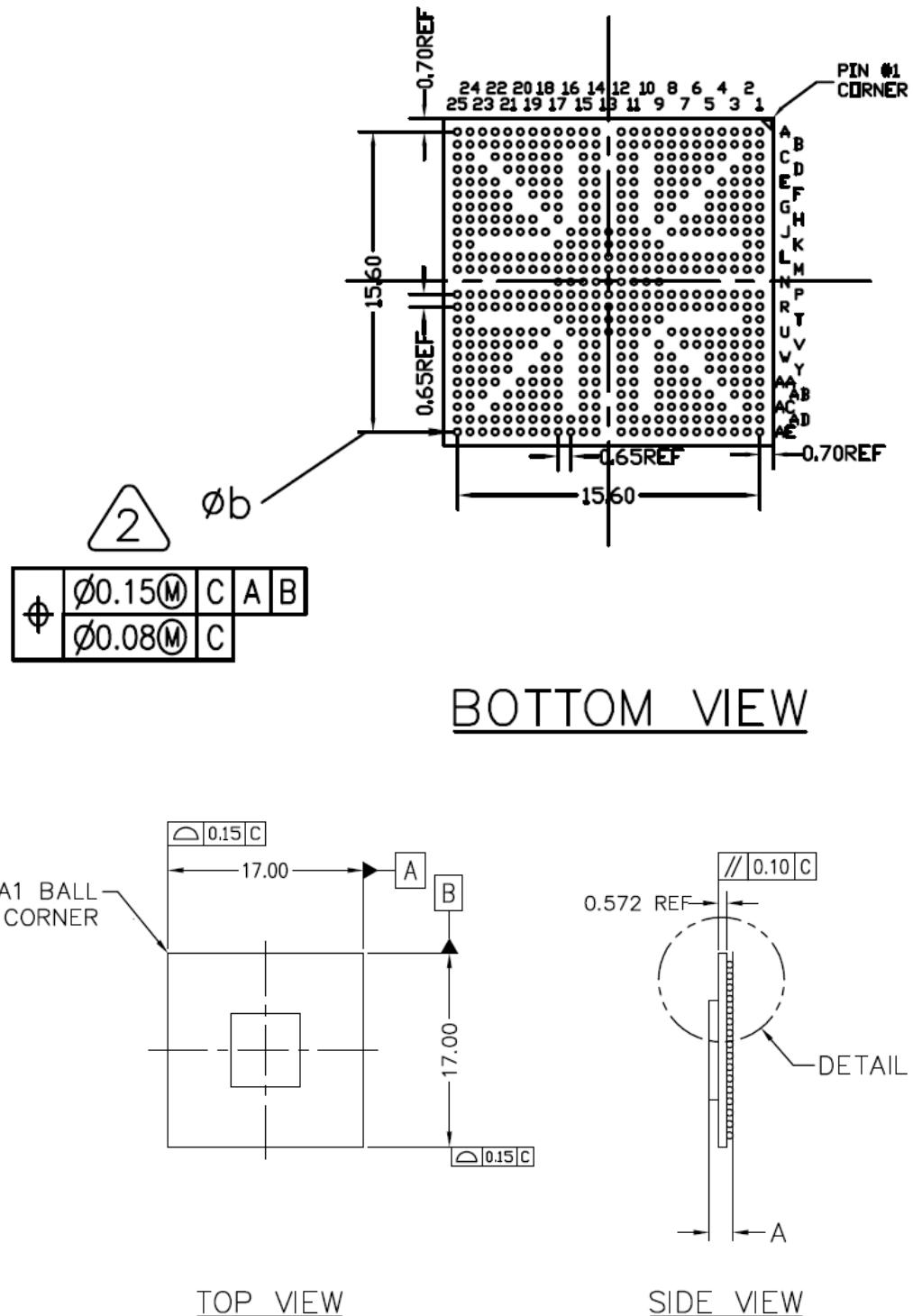
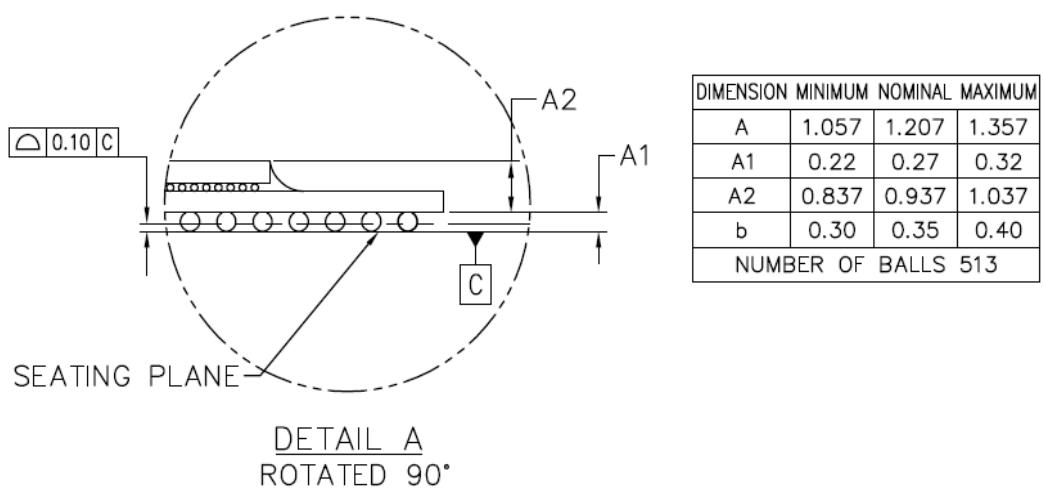


Figure 2-1. Mechanical Dimension - Top, Bottom, Side view



LIST OF MATERIAL AND APPLICABLE DOCUMENTS				
SCALE: N/S	DATE: 04/24/13	DRAWN:	TITLE: FCBGA 17X17MM 513 BALL 0.65MM BALL PITCH PACKAGE OUTLINE	
DIMENSIONAL UNIT: MM	UNTOLERANCED DIMENSIONS	ENGINEER:		
PROJECTION UNLESS SPECIFIED	FRAC: .X ±0.10 .XX ±0.05 .XXX ±0.03 ANGLE ±1°	CHECKED:		
		APPROVED:		

Figure 2-2. Mechanical Dimension - Dimension value

2.2 FCBGA Ball Map

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
A	MIPICSI_D_NCLK	MIPICSI_D_N0	MIPICSI_D_N1	MIPICSI_D_N2	MIPICSI_D_N3	VSSI	MIPIDSI_D_NCLK	MIPIDSI_D_N0	MIPIDSI_D_N1	MIPIDSI_D_N2	MIPIDSI_D_N3	GMAC_GT_XCLK		LVDS_TN1	LVDS_TN2	LVDS_TNCLK	LVDS_TN3	LVDS_TN4	VSSI	PLXTO	HDMI_REXT	HDMI_TXP2	HDMI_TXP1	HDMI_TXP0	HDMI_TXCLK	
B	MIPICSI_D_PCLK	MIPICSI_D_P0	MIPICSI_D_P1	MIPICSI_D_P2	MIPICSI_D_P3	VSSI	MIPIDSI_D_PCLK	MIPIDSI_D_P0	MIPIDSI_D_P1	MIPIDSI_D_P2	MIPIDSI_D_P3	GMAC_CR_S		LVDS_TP1	LVDS_TP2	LVDS_TPCLK	LVDS_TP3	LVDS_TP4	VSSI	PLXTO	HDMI_REXT	HDMI_TXP2	HDMI_TXP1	HDMI_TXP0	HDMI_TXCLK	
C	AD21	AD23		M_VDD10	M_VDD10	M_VDD10	M_VDD18	MIPIDSI_V_REG_0.4V	MIPIDSI_V_REG_0.4V	MIPIDSI_V_REG_0.4V	VSSI	GMAC_TX_D1	GMAC_TX_D2	LVDS_TN0	LVDS_TP0	GMAC_RX_D1	GMAC_RX_D3	LVDS_ROUT	VSS1B_O9_C	AVDD18_P PLL	AVDD10_H_M			USB2.0OT_G_VBUS_G_ID		
D	AD19	AD17	PADQ50		VDDI	VDDI	M_VDD10	M_VDD10	VSSI	GMAC_TX_D0	GMAC_TX_D3	GMAC_MD	GMAC_MD	GMAC_RX_D0	GMAC_RX_C	GMAC_RX_D0	GMAC_RX_D2	AVDD18_P	AVDD18_O9	AVDD18_P	AVDD18_HM	AVDD10_H	AVDD10_G_DM	USB2.0OT_G_DM		
E	ADQM2	PADQ52	AD2	NADQ50		VDDI	VDDI	VDDI	VSSI	GMAC_TXE_R	GMAC_TXE_N	GMAC_CO_L	GMAC_RX_ER	GMAC_RX_DV	GMAC_RX_CLK	AVSS1B_PL	AVSS1B_P_L	AVSS1B_P_LL	VDD33_USB0	VDD019_USB0	VDD019_USB0	VDD019_USB0	VDD019_USB0	VDD019_USB0		
F	NADQ52	AD20	ADQM0	AD6	AD0		VSSI	VSSI	VSSI	AVSS1B_LV	AVSS1B_LV	DVDD_GMAC	AVDD18_LV	AVSS1B_PLL	AVSS1B_PLL	VDD10_H_M_PLL	VDD18_USBHOST	VSSI	VDD18_US_B0	VDD18_US_ST_DM	VDD18_US_ST_DP					
G	AD16	AD22	AD3	AD4	AD1		VSSI	VSSI	VSSI	VDDI	VSSI	AVDD10_LV	AVSS10_LV	VSSI	AVSS1B_PLL	VDD10_H_M_PLL	VDD18_USBHOST	VSSI	VDD18_US_B0	VDD18_US_ST_DM	VDD18_US_ST_DP	VDD18_US_ST_RELV1_N	VDD33_USBHOST	VDD019_USBHOST		
H	AD18	ACK0	AD5	AD7	AA4	VSSI	VSSI		VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI	VSSI		
J	AA8	AA6	ABA1	AA1	ANWE	AA0	VSSI	VSSI		VSSI	VSSI	VDDI	VSSI	VDDI	VSSI	VDDI	VSSI	VSSI	VICI1	VIDI1_0	VISO19	VISO18	VISO11	VISO12		
K	AA14	AA11					VDDQ	VDDQ	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI		
L	ACKB	ACX	AA12	AA10	ABA2	VSSI	VSSI	VSSI	VDDQ	VDDQ	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI		
M	AA7	AA5	AA3	AA9	AA13	VSSI	VSSI	VSSI	VDDQ	VDDQ	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI	VDDI		
N						VDDQ	VDDQ	VDDI	VDDI	VDDI	VDDI	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM		
P	ANC50	ANCAS	ABA0	AA15	AODT1	VSSI	VSSI	VSSI	VDDQ	VDDQ	VDDI	VDDI	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_J_O	VID1_3	DISD22	DISD21	DISD20	DISD15	DISD0	
R	AODT0	ANRAS	AA2	ARST	ACKE1	VSSI	VSSI	VSSI	VDDQ	VDDQ	VDDI	VDDI	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_J_O	VID1_2	DISD23	DISD24	DISD31	DISD1	DISD14	
T	AREF1	AREF2					VDDQ	VDDQ	VDDI	VDDI	VDDI	VDDI	VDDI_VID_2_S02	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_J_O	VID1_4	DISD10	DISCLK	DISHSYNC	DISVSYNC	DISDE
U	AD8	AD10	AD27	AD25	ANCS1	VSSI	VSSI	VSSI	VDDI	VDDI	VDDI	VDDI	VDDI_VID_2_S02	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_ARM	VDDI_J_O	VID1_5	S06	S07	S0CM0	S0DATA1_I	S0DATA0_I
V	AD14	AD12	AD24	AD26	ZQ	NTRST	VDDP18_ALIVE		VDDP18	VDDP18	VDDI	VDDI		SA11	SA12	VDDI_J_O	VDDI_J_O	SA10	VID1_6	S05	S04	S03	S0DATA0_3	NNFWED		
W	ADOM1	PADQ51	NADQ53	NADQ53	TDI			VDD033_ALIVE	ALIVEGPIO_5		VDDI	VDDI		SA11	VSSI	UARTTXD3	VSSI		VID1_7	S02	S01	ALE0	CLE0			
Y	NADQ51	AD9	ADQM3	AD29	TMS		TDO	ALIVEGPIO_4	ALIVEGPIO_3		SA21	PPM		S0DAT1_3	S0DAT1_2	UARTRXD3	UARTTXD2	UARTRXD2		SA3	NNFOE0	S00	NNCS1	NNCS0		
AA	AD11	AD13	AD31	AD28		TCLK	ALIVEGPIO_2	ALIVEGPIO_1	VID0_0		VIHSYNC0	DVDD_VID0		SA17	I2SCLK0	S0DAT1_1	S0DAT1_0	S0CM01	S0CLK1		SA2	RNB0	NSDOM	S0B		
AB	AD15	NBATF	AD30		ADC1	VDDPWRO_N	NGRESETO_UT	ALIVEGPIO_0	ALIVEGPIO_4	VID0_4		SA20	SA18		SA19	I2SCLK0	SCL1	SDA1	SDA2	NSWE	NSCS1		RDNWR	S49	S09	
AC	ADC4	ADCREF		VDD18_RT_C	AVDD18_RT_ADC	VDDPWRO_N	VDD18_RT_C	VDD10_ALIVE	VID0_3		VID0_3	VID0_2	VICLK0	VIVSTNC0	SA14	SA22	I2SOUT0	SPIRXD0	SPIFRM0	UARTTXD1	UARTRXD0	NSCS0	S48	S08	SD15	SD11
AD	ADC5	ADC6	ADC2	RTCTO	AVSS18_ADC	ADC7	EFUSE_FS_OUTC	WIRED	VID0_2	VICLK0	VIVSTNC0	VID0_7	LATADDR		SA18	PWM0	SPITXD0	SPICLK0	UARTRXD1	UARTTXD0	NSOE	SA7	SA6	SA1	SD13	SD12
AE	NRESET	ADC3	ADC0	RTCTI	ADCREFGND	NVDOPWR_TOGGLE	TEST_EN	WIRE1	VID0_5	VID0_6	VID0_7	LATADDR														

Figure 2-3. FCBGA Ball Map (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	MIPICSI_D_NCLK	MIPICSI_DN0	MIPICSI_DN1	MIPICSI_DN2	MIPICSI_DN3	VSSI	MIPIDSI_D_NCLK	MIPIDSI_DN0	MIPIDSI_DN1	MIPIDSI_DN2	MIPIDSI_DN3	GMAC_GT_XCLK	
B	MIPICSI_DPCLK	MIPICSI_DP0	MIPICSI_DP1	MIPICSI_DP2	MIPICSI_DP3	VSSI	MIPIDSI_D_PCLK	MIPIDSI_D_P0	MIPIDSI_D_P1	MIPIDSI_D_P2	MIPIDSI_D_P3	GMAC_CR_S	
C	AD21	AD23		M_VDD10	M_VDD10	M_VDD18	MIPIDSI_V_REG_0P4V	M_VDD10_PLL	VSSI		GMAC_TX_D1	GMAC_TX_D2	
D	AD19	AD17	PADQSO		VDDI	VDDI	M_VDD10	M_VDD10	VSSI		GMAC_TX_D0	GMAC_TX_D3	
E	ADQM2	PADQS2	AD2	NADQS0		VDDI	VDDI	VDDI	VSSI		GMAC_TXE_R	GMAC_TXE_N	
F	NADQS2	AD20	ADQM0	AD6	AD0		VSSI	VSSI	VSSI		AVSS18_LV	AVSS18_LV	
G	AD16	AD22	AD3	AD4	AD1			VSSI	VSSI		VDDI	VSSI	
H	AD18	ACKE0	AD5	AD7	AA4	VSSI	VSSI		VSSI		VSSI	VSSI	
J	AA8	AA6	ABA1	AA1	ANWE	AA0	VSSI	VSSI		VSSI	VSSI	VDDI	VSSI
K	AA14	AA11							VDDQ	VDDQ	VDDI	VSSI	VDDI
L	ACKB	ACK	AA12	AA10	ABA2	VSSI	VSSI	VSSI	VDDQ	VDDQ	VDDI	VSSI	VSSI
M	AA7	AA5	AA3	AA9	AA13	VSSI	VSSI	VSSI	VDDQ	VDDQ	VDDI	VSSI	VDDI
N									VDDQ	VDDQ	VSSI	VDDI	VSSI

Figure 2-4. FCBGA Ball Map (Top View) - Upper Left side

	13	14	15	16	17	18	19	20	21	22	23	24	25	
A	LVDS_TN1	LVDS_TN2	LVDS_TNCLK	LVDS_TN3	LVDS_TN4	VSSI	PLLXTI	HDMI_HOT5V	HDMI_TXN2	HDMI_TXN1	HDMI_TXN0	HDMI_TXNCLK		
B	LVDS_TP1	LVDS_TP2	LVDS_TPCLK	LVDS_TP3	LVDS_TP4	VSSI	PLLXTO	HDMI_REXT	HDMI_TXP2	HDMI_TXP1	HDMI_TXP0	HDMI_TXPCLK		
C	LVDS_TN0	LVDS_TP0		GMAC_RX_D1	GMAC_RX_D3	LVDS_ROUT	VSS18_OS_C	AVDD18_PLL	AVDD10_HM		USB2.0OT_G_VBUS	USB2.0OT_G_ID		
D	GMAC_MD_IO	GMAC_MDC		GMAC_RX_D0	GMAC_RX_D2	AVDD18_PLL	VDD18_OS_C	AVDD18_PLL		VDD18_HM	USB2.0OT_G_DM	USB2.0OT_G_DP		
E	GMAC_CO_L	GMAC_RX_ER		GMAC_RX_DV	GMAC_RX_CLK	AVSS18_PLL	AVDD18_PLL		VDD33_USB0	DVDD10_USB0	USB2.0OT_G_USBVBU_S	USB2.0OT_G_RKELVIN		
F	DVDD_GMAC	AVDD18_LV		AVSS18_PLL	AVSS18_PLL	VDD10_HM_PLL		VDD18_USBHOST	VSSI	VDD18_USB0	USB2.0HO_ST_DM	USB2.0HO_ST_DP		
G	AVDD10_LV	AVSS10_LV		VSSI	AVSS18_PLL			VSSI	VSSI	DVDD10_USBHOST0	VDD33_USBHOST	USB2.0HO_ST_RKELVIN		
H	VSSI	VSSI		VSSI		VSSI	VSSI	DISD17	DISD16	DVDD12_HSIC	USBHSIC_DATA	USBHSIC_STROBE		
J	VSSI	VDDI	VSSI	VSSI		VSSI	VSSI	VCLK1	VID1_0	DISD19	DISD18	DISD11	DISD12	
K	VDDI	VSSI	VDDI	VSSI	VSSI							DISD13	DISD14	
L	VSSI	VDDI	VSSI	VDDI	VDDI	VSSI	DVDD33_I_O	VID1_1	DISD22	DISD21	DISD20	DISD15	DISD0	
M	VDDI	VSSI	VDDI	VSSI	VSSI	VSSI	DVDD33_I_O	VID1_2	DISD23	DISD4	DISD3	DISD1	DISD2	
N	VSSI	VDDI_ARM	VSSI	VDDI_ARM	VSSI									

Figure 2-5. FCBGA Ball Map (Top View) - Upper Right side

N								VDDQ	VDDQ	VSSI	VDDI	VSSI	
P	ANCS0	ANCAS	ABA0	AA15	AODT1	VSSI	VSSI	VDDQ	VDDQ	VDDI	VSSI	VDDI_ARM	
R	AODT0	ANRAS	AA2	ARST	ACKE1	VSSI	VSSI	VDDQ	VDDQ	VSSI	VDDI	VSSI	
T	AREF1	AREF2						VDDQ	VDDQ	VDDI	VSSI	DVDD_VID_2_SD2	
U	AD8	AD10	AD27	AD25	ANCS1	VSSI	VSSI	VSSI	VDDP18	VDDP18	VDDI	DVDD_VID_2_SD2	
V	AD14	AD12	AD24	AD26	ZQ	NTRST	VDDP18_A LIVE	VDDP18		VDDI	VDDI		
W	ADQM1	PADQS1	PADQS3	NADQS3	TDI		VDD33_ ALIVE	ALIVEGPIO 5		VDDI	VDDI		
Y	NADQS1	AD9	ADQM3	AD29	TMS		TDO	ALIVEGPIO 4	ALIVEGPIO 3	SA21	PPM		
AA	AD11	AD13	AD31	AD28		TCLK	ALIVEGPIO 2	ALIVEGPIO 1	VIDO_0		VIHSYNC0	DVDD_VID0	
AB	AD15	NBATF	AD30		ADC1	VDDPWRO_N	NGRESETOUT	ALIVEGPIO 0	VIDO_4		SA20	SA18	
AC	ADC4	ADCREF		VDD18_RT_C	AVDD18_ADC	VDDPWRO_N_DDR	VDD18_RT_C	VDDI10_ALIVE	VIDO_1		VIDO_3	SA23	
AD	ADC5	ADC6	ADC2	RTCXTO	AVSS18_ADC	ADC7	EFUSE_FS OURCE	WIRE0	VIDO_2	VICLKO	VIVSYNC0	SA14	
AE	NRESET	ADC3	ADC0	RTCTXI	ADCREFGN_D	NVDDPWR_TOGGLE	TEST_EN	WIRE1	VIDO_5	VIDO_6	VIDO_7	LATADDR	
	1	2	3	4	5	6	7	8	9	10	11	12	13

Figure 2-6. FCBGA Ball Map (Top View) - Lower Left side

VSSI	VDDI_ARM	VSSI	VDDI_ARM	VSSI											N
VDDI_ARM	VSSI	VDDI_ARM	VDDI_ARM	VSSI	VSSI	DVDD33_I_O	VID1_3	DISD9	DISD8	DISD5	DISD6	DISD7			P
VSSI	VDDI_ARM	VDDI_ARM	VDDI_ARM	VSSI	DVDD33_I_O	DVDD33_I_O	VID1_4	DISD10	DISCLK	DISHSYNC	DISVSYNC	DISDE			R
DVDD_VID_2_SD2	VDDI_ARM	VDDI_ARM	VDDI_ARM	VSSI									SDCLK0	SDDAT0_0	T
DVDD_VID_2_SD2	VDDI_ARM	VDDI_ARM	VSSI		DVDD33_I_O	DVDD33_I_O	VID1_5	SD6	SD7	SDCMD0	SDDAT0_1	SDDAT0_2			U
	SA13	SA12		DVDD33_I_O		SA10	VID1_6	SD5	SD4	SD3	SDDAT0_3	NNFWE0			V
	SA11	VSSI		UARTTXD3	VSSI			VID1_7	SD2	SD1	ALE0	CLE0			W
	SDDAT1_3	SDDAT1_2		UARTRXD3	UARTTXD2	UARTRXD2		SA3	NNFOE0	SD0	NNCS1	NNCS0			Y
	SA17	I2SMCLK0		SDDAT1_1	SDDAT1_0	SDCMD1	SDCLK1		SA2	RNB0	NSDQM	SD8			AA
	SA19	I2SBCLK0		SCL1	SDA1	SDA2	NSWE	NSCS1		RDNWR	SA9	SD9			AB
	SA22	I2SDIN0		I2SLRCLK0	SCL2	SDA0	SCL0	NSWAIT	SA4		SD15	SD10			AC
	SA15	I2SDOUT0	SPIRXDO	SPIFRM0	UARTTXD1	UARTTXD0	NSCS0	SA8	SA5	SA0	SD14	SD11			AD
	SA16	PWM0	SPITXDO	SPICLK0	UARTRXD1	UARTRXD0	NSOE	SA7	SA6	SA1	SD13	SD12			AE
13	14	15	16	17	18	19	20	21	22	23	24	25			

Figure 2-7. FCBGA Ball Map (Top View) - Lower Right side

2.3 I/O Function Description

2.3.1 Ball List Table

Note) Type definition - S: Signal ball, P: Power ball, G: GND ball

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
A1	MIPICSI_DNCLK	S	MIPICSI_DNCLK			
A2	MIPICSI_DN0	S	MIPICSI_DN0			
A3	MIPICSI_DN1	S	MIPICSI_DN1			
A4	MIPICSI_DN2	S	MIPICSI_DN2			
A5	MIPICSI_DN3	S	MIPICSI_DN3			
A6	VSSI	G				
A7	MIPIDSI_DNCLK	S	MIPIDSI_DNCLK			
A8	MIPIDSI_DN0	S	MIPIDSI_DN0			
A9	MIPIDSI_DN1	S	MIPIDSI_DN1			
A10	MIPIDSI_DN2	S	MIPIDSI_DN2			
A11	MIPIDSI_DN3	S	MIPIDSI_DN3			
A12	GMAC_GTXCLK	S	GPIOE24	GMAC_GTXCLK		
A14	LVDS_TN1	S	LVDS_TN1			
A15	LVDS_TN2	S	LVDS_TN2			
A16	LVDS_TNCLK	S	LVDS_TNCLK			
A17	LVDS_TN3	S	LVDS_TN3			
A18	LVDS_TN4	S	LVDS_TN4			
A19	VSSI	G				
A20	PLLXTI	S	PLLXTI			
A21	HDMI_HOT5V	S	HDMI_HOT5V			
A22	HDMI_TXN2	S	HDMI_TXN2			
A23	HDMI_TXN1	S	HDMI_TXN1			
A24	HDMI_TXN0	S	HDMI_TXN0			
A25	HDMI_TXNCLK	S	HDMI_TXNCLK			
B1	MIPICSI_DPCLK	S	MIPICSI_DPCLK			
B2	MIPICSI_DP0	S	MIPICSI_DP0			
B3	MIPICSI_DP1	S	MIPICSI_DP1			
B4	MIPICSI_DP2	S	MIPICSI_DP2			
B5	MIPICSI_DP3	S	MIPICSI_DP3			
B6	VSSI	G				
B7	MIPIDSI_DPCLK	S	MIPIDSI_DPCLK			
B8	MIPIDSI_DP0	S	MIPIDSI_DP0			

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
B 9	MIPIDSI_DP1	S	MIPIDSI_DP1			
B 10	MIPIDSI_DP2	S	MIPIDSI_DP2			
B 11	MIPIDSI_DP3	S	MIPIDSI_DP3			
B 12	GMAC_CRS	S	GPIOE23	GMAC_CRS		
B 14	LVDS_TP1	S	LVDS_TP1			
B 15	LVDS_TP2	S	LVDS_TP2			
B 16	LVDS_TPCLK	S	LVDS_TPCLK			
B 17	LVDS_TP3	S	LVDS_TP3			
B 18	LVDS_TP4	S	LVDS_TP4			
B 19	VSSI	G				
B 20	PLLXTO	S	PLLXTO			
B 21	HDMI_REXT	S	HDMI_REXT			
B 22	HDMI_TXP2	S	HDMI_TXP2			
B 23	HDMI_TXP1	S	HDMI_TXP1			
B 24	HDMI_TXP0	S	HDMI_TXP0			
B 25	HDMI_TXPCLK	S	HDMI_TXPCLK			
C1	AD21	S	AD21			
C2	AD23	S	AD23			
C4	M_VDD10	P				
C5	M_VDD10	P				
C6	M_VDD18	P				
C7	MIPIDSI_VREG_0P4V	S	MIPIDSI_VREG_0P4V			
C8	M_VDD10_PLL	P				
C9	VSSI	G				
C11	GMAC_TXD1	S	GPIOE8	GMAC_TXD1		
C12	GMAC_TXD2	S	GPIOE9	GMAC_TXD2		
C14	LVDS_TN0	S	LVDS_TN0			
C15	LVDS_TP0	S	LVDS_TP0			
C17	GMAC_RXD1	S	GPIOE15	GMAC_RXD1	SPIFRM1	
C18	GMAC_RXD3	S	GPIOE17	GMAC_RXD3		
C19	LVDS_ROUT	S	LVDS_ROUT			
C20	VSS18_OSC	G				
C21	AVDD18_PLL	P				
C22	AVDD10_HM	P				
C24	USB2.0OTG_VBUS	S	USB2.0OTG_VBUS			
C25	USB2.0OTG_ID	S	USB2.0OTG_ID			
D1	AD19	S	AD19			

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
D2	AD17	S	AD17			
D3	PADQS0	S	PADQS0			
D5	VDDI	P				
D6	VDDI	P				
D7	M_VDD10	P				
D8	M_VDD10	P				
D9	VSSI	G				
D11	GMAC_TXD0	S	GPIOE7	GMAC_TXD0	VIHSYNC1	
D12	GMAC_TXD3	S	GPIOE10	GMAC_TXD3		
D14	GMAC_MDIO	S	GPIOE21	GMAC_MDIO		
D15	GMAC_MDC	S	GPIOE20	GMAC_MDC		
D17	GMAC_RXD0	S	GPIOE14	GMAC_RXD0	SPICLK1	
D18	GMAC_RXD2	S	GPIOE16	GMAC_RXD2		
D19	AVDD18_PLL	P				
D20	VDD18_OSC	P				
D21	AVDD18_PLL	P				
D23	VDD18_HM	P				
D24	USB2.0OTG_DM	S	USB2.0OTG_DM			
D25	USB2.0OTG_DP	S	USB2.0OTG_DP			
E1	ADQM2	S	ADQM2			
E2	PADQS2	S	PADQS2			
E3	AD2	S	AD2			
E4	NADQS0	S	NADQS0			
E6	VDDI	P				
E7	VDDI	P				
E8	VDDI	P				
E9	VSSI	G				
E11	GMAC_TXER	S	GPIOE12	GMAC_TXER		
E12	GMAC_TXEN	S	GPIOE11	GMAC_TXEN		
E14	GMAC_COL	S	GPIOE13	GMAC_COL	VIHSYNC1	
E15	GMAC_RXER	S	GPIOE22	GMAC_RXER		
E17	GMAC_RXDV	S	GPIOE19	GMAC_RXDV	SPITXD1	
E18	GMAC_RXCLK	S	GPIOE18	GMAC_RXCLK	SPIRXD1	
E19	AVSS18_PLL	G				
E20	AVDD18_PLL	P				
E22	VDD33_USB0	P				
E23	DVDD10_USB0	P				

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
E 24	USB2.0OTG_USBVBUS	S	USB2.0OTG_USBVBUS			
E 25	USB2.0OTG_RKELVIN	S	USB2.0OTG_RKELVIN			
F1	NADQS2	S	NADQS2			
F2	AD20	S	AD20			
F3	ADQM0	S	ADQM0			
F4	AD6	S	AD6			
F5	AD0	S	AD0			
F7	VSSI	G				
F8	VSSI	G				
F9	VSSI	G				
F11	AVSS18_LV	G				
F12	AVSS18_LV	G				
F14	DVDD_GMAC	P				
F15	AVDD18_LV	P				
F17	AVSS18_PLL	G				
F18	AVSS18_PLL	G				
F19	VDD10_HM_PLL	P				
F21	VDD18_USBHOST	P				
F22	VSSI	G				
F23	VDD18_USB0	P				
F24	USB2.0HOST_DM	S	USB2.0HOST_DM			
F25	USB2.0HOST_DP	S	USB2.0HOST_DP			
G1	AD16	S	AD16			
G2	AD22	S	AD22			
G3	AD3	S	AD3			
G4	AD4	S	AD4			
G5	AD1	S	AD1			
G8	VSSI	G				
G9	VSSI	G				
G11	VDDI	P				
G12	VSSI	G				
G14	AVDD10_LV	P				
G15	AVSS10_LV	G				
G17	VSSI	G				
G18	AVSS18_PLL	G				
G21	VSSI	G				
G22	VSSI	G				

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
G 23	DVDD10_USBHOST0	P				
G 24	VDD33_USBHOST	P				
G 25	USB2.0HOST_RKELVIN	S	USB2.0HOST_RKELVIN			
H1	AD18	S	AD18			
H2	ACKE0	S	ACKE0			
H3	AD5	S	AD5			
H4	AD7	S	AD7			
H5	AA4	S	AA4			
H6	VSSI	G				
H7	VSSI	G				
H9	VSSI	G				
H11	VSSI	G				
H12	VSSI	G				
H14	VSSI	G				
H15	VSSI	G				
H17	VSSI	G				
H19	VSSI	G				
H20	VSSI	G				
H21	DISD17	S	GPIOA18	DISD17		
H22	DISD16	S	GPIOA17	DISD16		
H23	DVDD12_HSIC	P				
H24	USBHSIC_DATA	S	USBHSIC_DATA			
H25	USBHSIC_STROBE	S	USBHSIC_STROBE			
J1	AA8	S	AA8			
J2	AA6	S	AA6			
J3	ABA1	S	ABA1			
J4	AA1	S	AA1			
J5	ANWE	S	ANWE			
J6	AA0	S	AA0			
J7	VSSI	G				
J8	VSSI	G				
J10	VSSI	G				
J11	VSSI	G				
J12	VDDI	P				
J13	VSSI	G				
J14	VDDI	P				
J15	VSSI	G				

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
J16	VSSI	G				
J18	VSSI	G				
J19	VSSI	G				
J20	VICLK1	S	GPIOA28	VICLK1	I2SMCLK2	I2SMCLK1
J21	VID1_0	S	GPIOA30	VID1_0	SDEX0	I2SBCLK1
J22	DISD19	S	GPIOA20	DISD19		
J23	DISD18	S	GPIOA19	DISD18		
J24	DISD11	S	GPIOA12	DISD11		
J25	DISD12	S	GPIOA13	DISD12		
K1	AA14	S	AA14			
K2	AA11	S	AA11			
K9	VDDQ	P				
K10	VDDQ	P				
K11	VDDI	P				
K12	VSSI	G				
K13	VDDI	P				
K14	VSSI	G				
K15	VDDI	P				
K16	VSSI	G				
K17	VSSI	G				
K24	DISD13	S	GPIOA14	DISD13		
K25	DISD14	S	GPIOA15	DISD14		
L1	ACKB	S	ACKB			
L2	ACK	S	ACK			
L3	AA12	S	AA12			
L4	AA10	S	AA10			
L5	ABA2	S	ABA2			
L6	VSSI	G				
L7	VSSI	G				
L8	VSSI	G				
L9	VDDQ	P				
L10	VDDQ	P				
L11	VSSI	G				
L12	VDDI	P				
L13	VSSI	G				
L14	VDDI	P				
L15	VSSI	G				

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
L 16	VDDI	P				
L 17	VDDI	P				
L 18	VSSI	G				
L 19	DVDD33_IO	P				
L 20	VID1_1	S	GPIOB0	VID1_1	SDEX1	I2SLRCLK1
L 21	DISD22	S	GPIOA23	DISD22		
L 22	DISD21	S	GPIOA22	DISD21		
L 23	DISD20	S	GPIOA21	DISD20		
L 24	DISD15	S	GPIOA16	DISD15		
L 25	DISD0	S	GPIOA1	DISD0		
M1	AA7	S	AA7			
M2	AA5	S	AA5			
M3	AA3	S	AA3			
M4	AA9	S	AA9			
M5	AA13	S	AA13			
M6	VSSI	G				
M7	VSSI	G				
M8	VSSI	G				
M9	VDDQ	P				
M10	VDDQ	P				
M11	VDDI	P				
M12	VSSI	G				
M13	VDDI	P				
M14	VSSI	G				
M15	VDDI	P				
M16	VSSI	G				
M17	VSSI	G				
M18	VSSI	G				
M19	DVDD33_IO	P				
M20	VID1_2	S	GPIOB2	VID1_2	SDEX2	I2SBCLK2
M21	DISD23	S	GPIOA24	DISD23		
M22	DISD4	S	GPIOA5	DISD4		
M23	DISD3	S	GPIOA4	DISD3		
M24	DISD1	S	GPIOA2	DISD1		
M25	DISD2	S	GPIOA3	DISD2		
N 9	VDDQ	P				
N 10	VDDQ	P				

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
N 11	VSSI	G				
N 12	VDDI	P				
N 13	VSSI	G				
N 14	VDDI_ARM	P				
N 15	VSSI	G				
N 16	VDDI_ARM	P				
N 17	VSSI	G				
P1	ANCS0	S	ANCS0			
P2	ANCAS	S	ANCAS			
P3	ABA0	S	ABA0			
P4	AA15	S	AA15			
P5	AODT1	S	AODT1			
P6	VSSI	G				
P7	VSSI	G				
P8	VSSI	G				
P9	VDDQ	P				
P10	VDDQ	P				
P11	VDDI	P				
P12	VSSI	G				
P13	VDDI_ARM	P				
P14	VSSI	G				
P15	VDDI_ARM	P				
P16	VDDI_ARM	P				
P17	VSSI	G				
P18	VSSI	G				
P19	DVDD33_IO	P				
P20	VID1_3	S	GPIOB4	VID1_3	SDEX3	I2SLRCLK2
P21	DISD9	S	GPIOA10	DISD9		
P22	DISD8	S	GPIOA9	DISD8		
P23	DISD5	S	GPIOA6	DISD5		
P24	DISD6	S	GPIOA7	DISD6		
P25	DISD7	S	GPIOA8	DISD7		
R1	AODT0	S	AODT0			
R2	ANRAS	S	ANRAS			
R3	AA2	S	AA2			
R4	ARST	S	ARST			
R5	ACKE1	S	ACKE1			

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
R 6	VSSI	G				
R 7	VSSI	G				
R 8	VSSI	G				
R 9	VDDQ	P				
R 10	VDDQ	P				
R 11	VSSI	G				
R 12	VDDI	P				
R 13	VSSI	G				
R 14	VDDI_ARM	P				
R 15	VDDI_ARM	P				
R 16	VDDI_ARM	P				
R 17	VSSI	G				
R 18	DVDD33_IO	P				
R 19	DVDD33_IO	P				
R 20	VID1_4	S	GPIOB6	VID1_4	SDEX4	I2SDOUT1
R 21	DISD10	S	GPIOA11	DISD10		
R 22	DISCLK	S	GPIOA0	DISCLK		
R 23	DISHSYNC	S	GPIOA26	DISHSYNC		
R 24	DISVSYNC	S	GPIOA25	DISVSYNC		
R 25	DISDE	S	GPIOA27	DISDE		
T1	AREF1	P	AREF1			
T2	AREF2	P	AREF2			
T9	VDDQ	P				
T10	VDDQ	P				
T11	VDDI	P				
T12	VSSI	G				
T13	DVDD_VID2_SD2	P				
T14	VDDI_ARM	P				
T15	VDDI_ARM	P				
T16	VDDI_ARM	P				
T17	VSSI	G				
T24	SDCLK0	S	GPIOA29	SDCLK0		
T25	SDDAT0_0	S	GPIOB1	SDDAT0_0		
U 1	AD8	S	AD8			
U 2	AD10	S	AD10			
U 3	AD27	S	AD27			
U 4	AD25	S	AD25			

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
U5	ANCS1	S	ANCS1			
U6	VSSI	G				
U7	VSSI	G				
U8	VSSI	G				
U10	VDDP18	P				
U11	VDDP18	P				
U12	VDDI	P				
U13	DVDD_VID2_SD2	P				
U14	VDDI_ARM	P				
U15	VDDI_ARM	P				
U16	VSSI	G				
U18	DVDD33_IO	P				
U19	DVDD33_IO	P				
U20	VID1_5	S	GPIOB8	VID1_5	SDEX5	I2SDOUT2
U21	SD6	S	SD6	GPIOB22		
U22	SD7	S	SD7	GPIOB23		
U23	SDCMD0	S	GPIOA31	SDCMD0		
U24	SDDAT0_1	S	GPIOB3	SDDAT0_1		
U25	SDDAT0_2	S	GPIOB5	SDDAT0_2		
V1	AD14	S	AD14			
V2	AD12	S	AD12			
V3	AD24	S	AD24			
V4	AD26	S	AD26			
V5	ZQ	S	ZQ			
V6	NTRST	S	NTRST	GPIOE25		
V7	VDDP18_ALIVE	P				
V9	VDDP18	P				
V11	VDDI	P				
V12	VDDI	P				
V14	SA13	S	SA13	GPIOC13	PWM1	SDnINT2
V15	SA12	S	SA12	GPIOC12	SPITXD2	SDnRST2
V17	DVDD33_IO	P				
V19	SA10	S	SA10	GPIOC10	SPIFRM2	
V20	VID1_6	S	GPIOB9	VID1_6	SDEX6	I2SDIN1
V21	SD5	S	SD5	GPIOB21		
V22	SD4	S	SD4	GPIOB20		
V23	SD3	S	SD3	GPIOB19		

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
V24	SDDAT0_3	S	GPIOB7	SDDAT0_3		
V25	NNFWE0	S	NNFWE0	nNFWE1	GPIOB18	
W1	ADQM1	S	ADQM1			
W2	PADQS1	S	PADQS1			
W3	PADQS3	S	PADQS3			
W4	NADQS3	S	NADQS3			
W5	TDI	S	TDI	GPIOE27		
W8	VDD33_ALIVE	P				
W9	ALIVEGPIO5	S	ALIVEGPIO5			
W11	VDDI	P				
W12	VDDI	P				
W14	SA11	S	SA11	GPIOC11	SPIRXD2	USB2.0OTG_DrvBUS
W15	VSSI	G				
W17	UARTTXD3	S	GPIOD21	UARTTXD3	CANTX1	SDnCD1
W18	VSSI	G				
W21	VID1_7	S	GPIOB10	VID1_7	SDEX7	I2SDIN2
W22	SD2	S	SD2	GPIOB17		
W23	SD1	S	SD1	GPIOB15		
W24	ALE0	S	ALE0	ALE1	GPIOB12	
W25	CLE0	S	CLE0	CLE1	GPIOB11	
Y1	NADQS1	S	NADQS1			
Y2	AD9	S	AD9			
Y3	ADQM3	S	ADQM3			
Y4	AD29	S	AD29			
Y5	TMS	S	TMS	GPIOE26		
Y7	TDO	S	TDO	GPIOE29		
Y8	ALIVEGPIO4	S	ALIVEGPIO4			
Y9	ALIVEGPIO3	S	ALIVEGPIO3			
Y11	SA21	S	SA21	GPIOC21	SDDAT2_1	VID2_4
Y12	PPM	S	GPIOD8	PPM		
Y14	SDDAT1_3	S	GPIOD27	SDDAT1_3		
Y15	SDDAT1_2	S	GPIOD26	SDDAT1_2		
Y17	UARTRXD3	S	GPIOD17	UARTRXD3	CANRX1	
Y18	UARTTXD2	S	GPIOD20	UARTTXD2	CANTX0	SDWP1
Y19	UARTRXD2	S	GPIOD16	UARTRXD2	CANRX0	
Y21	SA3	S	SA3	GPIOC3	HDMI_CEC	SDnRST0
Y22	NNFOE0	S	NNFOE0	NNFOE1	GPIOB16	

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
Y23	SD0	S	SD0	GPIOB13		
Y24	NNCS1	S	NNCS1			
Y25	NNCS0	S	NNCS0			
AA1	AD11	S	AD11			
AA2	AD13	S	AD13			
AA3	AD31	S	AD31			
AA4	AD28	S	AD28			
AA6	TCLK	S	TCLK	GPIOE28		
AA7	ALIVEGPIO2	S	ALIVEGPIO2			
AA8	ALIVEGPIO1	S	ALIVEGPIO1			
AA9	VID0_0	S	GPIOD28	VID0_0	TSIDATA1_0	SA24
AA11	VIHSYNC0	S	GPIOE5	VIHSYNC0	TSISYNC1	
AA12	DVDD_VID0	P				
AA14	SA17	S	SA17	GPIOC17	TSIDP0	VID2_0
AA15	I2SMCLK0	S	GPIOD13	I2SMCLK0	AC97_nRST	
AA17	SDDAT1_1	S	GPIOD25	SDDAT1_1		
AA18	SDDAT1_0	S	GPIOD24	SDDAT1_0		
AA19	SDCMD1	S	GPIOD23	SDCMD1		
AA20	SDCLK1	S	GPIOD22	SDCLK1		
AA22	SA2	S	SA2	GPIOC2		
AA23	RNB0	S	RnB0	RnB1	GPIOB14	
AA24	NSDQM	S	NSDQM	GPIOC27	PDMDATA1	
AA25	SD8	S	SD8	GPIOB24	TSIDATA0_0	
AB1	AD15	S	AD15			
AB2	NBATF	S	NBATF			
AB3	AD30	S	AD30			
AB5	ADC1	S	ADC1			
AB6	VDDPWRON	S	VDDPWRON			
AB7	NGRESETOUT	S	NGRESETOUT			
AB8	ALIVEGPIO0	S	ALIVEGPIO0			
AB9	VID0_4	S	GPIOE0	VID0_4	TSIDATA1_4	
AB11	SA20	S	SA20	GPIOC20	SDDAT2_0	VID2_3
AB12	SA18	S	SA18	GPIOC18	SDCLK2	VID2_1
AB14	SA19	S	SA19	GPIOC19	SDCMD2	VID2_2
AB15	I2SBCLK0	S	GPIOD10	I2SBCLK0	AC97_BCLK	
AB17	SCL1	S	GPIOD4	SCL1		
AB18	SDA1	S	GPIOD5	SDA1		

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AB 19	SDA2	S	GPIOD7	SDA2		
AB 20	NSWE	S	NSWE	GPIOE31		
AB 21	NSCS1	S	GPIOC28	NSCS1	UARTnR1I	
AB 23	RDNWR	S	RDNWR	GPIOC26	PDM DATA0	
AB 24	SA9	S	SA9	GPIOC9	SPICLK2	PDM Strobe
AB 25	SD9	S	SD9	GPIOB25	TSIDATA0_1	
AC1	ADC4	S	ADC4			
AC2	ADCREF	P				
AC4	VDD18_RTC	P				
AC5	AVDD18_ADC	P				
AC6	VDDPWRON_DDR	S	VDDPWRON_DDR			
AC7	VDD18_RTC	P				
AC8	VDDI10_ALIVE	P				
AC9	VID0_1	S	GPIOD29	VID0_1	TSIDATA1_1	
AC11	VID0_3	S	GPIOD31	VID0_3	TSIDATA1_3	
AC12	SA23	S	SA23	GPIOC23	SDDAT2_3	VID2_6
AC14	SA22	S	SA22	GPIOC22	SDDAT2_2	VID2_5
AC15	I2SDIN0	S	GPIOD11	I2SDIN0	AC97_DIN	
AC17	I2SLRCLK0	S	GPIOD12	I2SLRCLK0	AC97_SYNC	
AC18	SCL2	S	GPIOD6	SCL2		
AC19	SDA0	S	GPIOD3	SDA0	ISO7816	
AC20	SCL0	S	GPIOD2	SCL0	ISO7816	
AC21	NSWAIT	S	NSWAIT	GPIOC25	SPDIFTX	
AC22	SA4	S	SA4	GPIOC4	UARTnDCD1	SDnINT0
AC24	SD15	S	SD15	GPIOB31	TSIDATA0_7	UARTTXD5
AC25	SD10	S	SD10	GPIOB26	TSIDATA0_2	
AD1	ADC5	S	ADC5			
AD2	ADC6	S	ADC6			
AD3	ADC2	S	ADC2			
AD4	RTCXTO	S	RTCXTO			
AD5	AVSS18_ADC	G				
AD6	ADC7	S	ADC7			
AD7	EFUSE_FSOURCE	S	EFUSE_FSOURCE			
AD8	WIRE0	S	WIRE0			
AD9	VID0_2	S	GPIOD30	VID0_2	TSIDATA1_2	
AD10	VICLK0	S	GPIOE4	VICLK0	TSICLK1	
AD11	VIVSYNC0	S	GPIOE6	VIVSYNC0	TSIDP1	

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AD 12	SA14	S	SA14	GPIOC14	PWM2	VICLK2
AD 14	SA15	S	SA15	GPIOC15	TSICLK0	VIHsync2
AD 15	I2SDOUT0	S	GPIOD9	I2SDOUT0	AC97_DOUT	
AD 16	SPIRXD0	S	GPIOD0	SPIRXD0	PWM3	
AD 17	SPIFRM0	S	GPIOC30	SPIFRM0		
AD 18	UARTTXD1	S	GPIOD19	UARTTXD1	ISO7816	SDnCD2
AD 19	UARTTXD0	S	GPIOD18	UARTTXD0	ISO7816	SDWP2
AD 20	NSCS0	S	NSCS0			
AD 21	SA8	S	SA8	GPIOC8	UARTnDTR1	SDnINT1
AD 22	SA5	S	SA5	GPIOC5	UARTnCTS1	SDWP0
AD 23	SA0	S	SA0	GPIOC0		
AD 24	SD14	S	SD14	GPIOB30	TSIDATA0_6	UARTRXD5
AD 25	SD11	S	SD11	GPIOB27	TSIDATA0_3	
AE1	NRESET	S	NRESET			
AE2	ADC3	S	ADC3			
AE3	ADC0	S	ADC0			
AE4	RTCXTI	S	RTCXTI			
AE5	ADCREFGND	G				
AE6	NVDDPWRTOGGLE	S	NVDDPWRTOGGLE			
AE7	TEST_EN	S	TEST_EN			
AE8	WIRE1	S	WIRE1			
AE9	VID0_5	S	GPIOE1	VID0_5	TSIDATA1_5	
AE10	VID0_6	S	GPIOE2	VID0_6	TSIDATA1_6	
AE11	VID0_7	S	GPIOE3	VID0_7	TSIDATA1_7	
AE12	LATADDR	S	LATADDR	GPIOC24	SPDIFRX	VID2_7
AE14	SA16	S	SA16	GPIOC16	TSISYNC0	VIVSYNC2
AE15	PWM0	S	GPIOD1	PWM0	SA25	
AE16	SPITXD0	S	GPIOC31	SPITXD0		
AE17	SPICLK0	S	GPIOC29	SPICLK0		
AE18	UARTRXD1	S	GPIOD15	UARTRXD1	ISO7816	
AE19	UARTRXD0	S	GPIOD14	UARTRXD0	ISO7816	
AE20	NSOE	S	NSOE	GPIOE30		
AE21	SA7	S	SA7	GPIOC7	UARTnDSR1	SDnRST1
AE22	SA6	S	SA6	GPIOC6	UARTnRTS1	SDnCD0
AE23	SA1	S	SA1	GPIOC1		
AE24	SD13	S	SD13	GPIOB29	TSIDATA0_5	UARTTXD4
AE25	SD12	S	SD12	GPIOB28	TSIDATA0_4	UARTRXD4

Table 2-1. Ball Function Table

2.3.2 Ball List Table: sorted by Function

2.3.2.1 MCU-A

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
F5	AD0	S	AD0			
G5	AD1	S	AD1			
E3	AD2	S	AD2			
G3	AD3	S	AD3			
G4	AD4	S	AD4			
H3	AD5	S	AD5			
F4	AD6	S	AD6			
H4	AD7	S	AD7			
U1	AD8	S	AD8			
Y2	AD9	S	AD9			
U2	AD10	S	AD10			
AA1	AD11	S	AD11			
V2	AD12	S	AD12			
AA2	AD13	S	AD13			
V1	AD14	S	AD14			
AB1	AD15	S	AD15			
G1	AD16	S	AD16			
D2	AD17	S	AD17			
H1	AD18	S	AD18			
D1	AD19	S	AD19			
F2	AD20	S	AD20			
C1	AD21	S	AD21			
G2	AD22	S	AD22			
C2	AD23	S	AD23			
V3	AD24	S	AD24			
U4	AD25	S	AD25			
V4	AD26	S	AD26			
U3	AD27	S	AD27			
AA4	AD28	S	AD28			
Y4	AD29	S	AD29			
AB3	AD30	S	AD30			
AA3	AD31	S	AD31			
J6	AA0	S	AA0			
J4	AA1	S	AA1			

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
R3	AA2	S	AA2			
M3	AA3	S	AA3			
H5	AA4	S	AA4			
M2	AA5	S	AA5			
J2	AA6	S	AA6			
M1	AA7	S	AA7			
J1	AA8	S	AA8			
M4	AA9	S	AA9			
L4	AA10	S	AA10			
K2	AA11	S	AA11			
L3	AA12	S	AA12			
M5	AA13	S	AA13			
K1	AA14	S	AA14			
P4	AA15	S	AA15			
P3	ABA0	S	ABA0			
J3	ABA1	S	ABA1			
L5	ABA2	S	ABA2			
D3	PADQS0	S	PADQS0			
W2	PADQS1	S	PADQS1			
E2	PADQS2	S	PADQS2			
W3	PADQS3	S	PADQS3			
E4	NADQS0	S	NADQS0			
Y1	NADQS1	S	NADQS1			
F1	NADQS2	S	NADQS2			
W4	NADQS3	S	NADQS3			
F3	ADQM0	S	ADQM0			
W1	ADQM1	S	ADQM1			
E1	ADQM2	S	ADQM2			
Y3	ADQM3	S	ADQM3			
R2	ANRAS	S	ANRAS			
P2	ANCAS	S	ANCAS			
P1	ANCS0	S	ANCS0			
U5	ANCS1	S	ANCS1			
L2	ACK	S	ACK			
L1	ACKB	S	ACKB			
H2	ACKE0	S	ACKE0			
R5	ACKE1	S	ACKE1			

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
J5	ANWE	S	ANWE			
R4	ARST	S	ARST			
R1	AODT0	S	AODT0			
P5	AODT1	S	AODT1			
T1	AREF1	P	AREF1			
T2	AREF2	P	AREF2			
V5	ZQ	S	ZQ			

2.3.2.2 MCU-S

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
Y23	SD0	S	SD0	GPIOB13		
W23	SD1	S	SD1	GPIOB15		
W22	SD2	S	SD2	GPIOB17		
V23	SD3	S	SD3	GPIOB19		
V22	SD4	S	SD4	GPIOB20		
V21	SD5	S	SD5	GPIOB21		
U21	SD6	S	SD6	GPIOB22		
U22	SD7	S	SD7	GPIOB23		
AA25	SD8	S	SD8	GPIOB24	TSIDATA0_0	
AB25	SD9	S	SD9	GPIOB25	TSIDATA0_1	
AC25	SD10	S	SD10	GPIOB26	TSIDATA0_2	
AD25	SD11	S	SD11	GPIOB27	TSIDATA0_3	
AE25	SD12	S	SD12	GPIOB28	TSIDATA0_4	UARTRXD4
AE24	SD13	S	SD13	GPIOB29	TSIDATA0_5	UARTTXD4
AD24	SD14	S	SD14	GPIOB30	TSIDATA0_6	UARTRXD5
AC24	SD15	S	SD15	GPIOB31	TSIDATA0_7	UARTTXD5
J21	SDEX0	S	GPIOA30	VID1_0	SDEX0	I2SBCLK1
L20	SDEX1	S	GPIOB0	VID1_1	SDEX1	I2SLRCLK1
M20	SDEX2	S	GPIOB2	VID1_2	SDEX2	I2SBCLK2
P20	SDEX3	S	GPIOB4	VID1_3	SDEX3	I2SLRCLK2
R20	SDEX4	S	GPIOB6	VID1_4	SDEX4	I2SDOUT1
U20	SDEX5	S	GPIOB8	VID1_5	SDEX5	I2SDOUT2
V20	SDEX6	S	GPIOB9	VID1_6	SDEX6	I2SDIN1
W21	SDEX7	S	GPIOB10	VID1_7	SDEX7	I2SDIN2
AD23	SA0	S	SA0	GPIOC0		
AE23	SA1	S	SA1	GPIOC1		

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AA 22	SA2	S	SA2	GPIOC2		
Y 21	SA3	S	SA3	GPIOC3	HDMI_CEC	SDnRST0
AC 22	SA4	S	SA4	GPIOC4	UARTnDCD1	SDnINT0
AD 22	SA5	S	SA5	GPIOC5	UARTnCTS1	SDWP0
AE 22	SA6	S	SA6	GPIOC6	UARTnRTS1	SDnCD0
AE 21	SA7	S	SA7	GPIOC7	UARTnDSR1	SDnRST1
AD 21	SA8	S	SA8	GPIOC8	UARTnDTR1	SDnINT1
AB 24	SA9	S	SA9	GPIOC9	SPICLK2	PDMStrobe
V 19	SA10	S	SA10	GPIOC10	SPIFRM2	
W 14	SA11	S	SA11	GPIOC11	SPIRXD2	USB2.0OTG_DnVBUS
V 15	SA12	S	SA12	GPIOC12	SPITXD2	SDnRST2
V 14	SA13	S	SA13	GPIOC13	PWM1	SDnINT2
AD 12	SA14	S	SA14	GPIOC14	PWM2	VICLK2
AD 14	SA15	S	SA15	GPIOC15	TSICLK0	VIHsync2
AE 14	SA16	S	SA16	GPIOC16	TSISYNC0	VIVSYNC2
AA 14	SA17	S	SA17	GPIOC17	TSIDP0	VID2_0
AB 12	SA18	S	SA18	GPIOC18	SDCLK2	VID2_1
AB 14	SA19	S	SA19	GPIOC19	SDCMD2	VID2_2
AB 11	SA20	S	SA20	GPIOC20	SDDAT2_0	VID2_3
Y 11	SA21	S	SA21	GPIOC21	SDDAT2_1	VID2_4
AC 14	SA22	S	SA22	GPIOC22	SDDAT2_2	VID2_5
AC 12	SA23	S	SA23	GPIOC23	SDDAT2_3	VID2_6
AA 9	SA24	S	GPIOD28	VID0_0	TSIDATA1_0	SA24
AE 15	SA25	S	GPIOD1	PWM0	SA25	
AD 20	NSCS0	S	NSCS0			
AB 21	NSCS1	S	GPIOC28	NSCS1	UARTnR11	
AB 20	NSWE	S	NSWE	GPIOE31		
AE 20	NSOE	S	NSOE	GPIOE30		
AB 23	RDNWR	S	RDNWR	GPIOC26	PDMDATA0	
AA 24	NSDQM	S	NSDQM	GPIOC27	PDMDATA1	
AC 21	NSWAIT	S	NSWAIT	GPIOC25	SPDIFTX	
AE 12	LATADDR	S	LATADDR	GPIOC24	SPDIFRX	VID2_7
W 25	CLE0	S	CLE0	CLE1	GPIOB11	
W 24	ALE0	S	ALE0	ALE1	GPIOB12	
AA 23	RnB0	S	RnB0	RnB1	GPIOB14	
Y 22	NNFOE0	S	NNFOE0	NNFOE1	GPIOB16	
V 25	NNFWE0	S	NNFWE0	nNFWE1	GPIOB18	

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
Y25	NNCS0	S	NNCS0			
Y24	NNCS1	S	NNCS1			

2.3.2.3 Digital RGB

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
L25	DISD0	S	GPIOA1	DISD0		
M24	DISD1	S	GPIOA2	DISD1		
M25	DISD2	S	GPIOA3	DISD2		
M23	DISD3	S	GPIOA4	DISD3		
M22	DISD4	S	GPIOA5	DISD4		
P23	DISD5	S	GPIOA6	DISD5		
P24	DISD6	S	GPIOA7	DISD6		
P25	DISD7	S	GPIOA8	DISD7		
P22	DISD8	S	GPIOA9	DISD8		
P21	DISD9	S	GPIOA10	DISD9		
R21	DISD10	S	GPIOA11	DISD10		
J24	DISD11	S	GPIOA12	DISD11		
J25	DISD12	S	GPIOA13	DISD12		
K24	DISD13	S	GPIOA14	DISD13		
K25	DISD14	S	GPIOA15	DISD14		
L24	DISD15	S	GPIOA16	DISD15		
H22	DISD16	S	GPIOA17	DISD16		
H21	DISD17	S	GPIOA18	DISD17		
J23	DISD18	S	GPIOA19	DISD18		
J22	DISD19	S	GPIOA20	DISD19		
L23	DISD20	S	GPIOA21	DISD20		
L22	DISD21	S	GPIOA22	DISD21		
L21	DISD22	S	GPIOA23	DISD22		
M21	DISD23	S	GPIOA24	DISD23		
R22	DISCLK	S	GPIOA0	DISCLK		
R24	DISVSYNC	S	GPIOA25	DISVSYNC		
R23	DISHSYNC	S	GPIOA26	DISHSYNC		
R25	DISDE	S	GPIOA27	DISDE		

2.3.2.4 HDMI

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
B24	HDMI_TXP0	S	HDMI_TXP0			
A24	HDMI_TXN0	S	HDMI_TXN0			
B23	HDMI_TXP1	S	HDMI_TXP1			
A23	HDMI_TXN1	S	HDMI_TXN1			
B22	HDMI_TXP2	S	HDMI_TXP2			
A22	HDMI_TXN2	S	HDMI_TXN2			
B25	HDMI_TXPCLK	S	HDMI_TXPCLK			
A25	HDMI_TXNCLK	S	HDMI_TXNCLK			
Y21	HDMI_CEC	S	SA3	GPIOC3	HDMI_CEC	SDnRST0
A21	HDMI_HOT5V	S	HDMI_HOT5V			
B21	HDMI_RECT	S	HDMI_RECT			

2.3.2.5 LVDS

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
C15	LVDS_TP0	S	LVDS_TP0			
C14	LVDS_TN0	S	LVDS_TN0			
B14	LVDS_TP1	S	LVDS_TP1			
A14	LVDS_TN1	S	LVDS_TN1			
B15	LVDS_TP2	S	LVDS_TP2			
A15	LVDS_TN2	S	LVDS_TN2			
B17	LVDS_TP3	S	LVDS_TP3			
A17	LVDS_TN3	S	LVDS_TN3			
B18	LVDS_TP4	S	LVDS_TP4			
A18	LVDS_TN4	S	LVDS_TN4			
B16	LVDS_TPCLK	S	LVDS_TPCLK			
A16	LVDS_TNCLK	S	LVDS_TNCLK			
C19	LVDS_ROUT	S	LVDS_ROUT			

2.3.2.6 MIPI DSI

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
B8	MIPIDSI_DP0	S	MIPIDSI_DP0			
A8	MIPIDSI_DN0	S	MIPIDSI_DN0			
B9	MIPIDSI_DP1	S	MIPIDSI_DP1			
A9	MIPIDSI_DN1	S	MIPIDSI_DN1			

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
B 10	MIPIDSI_DP2	S	MIPIDSI_DP2			
A 10	MIPIDSI_DN2	S	MIPIDSI_DN2			
B 11	MIPIDSI_DP3	S	MIPIDSI_DP3			
A 11	MIPIDSI_DN3	S	MIPIDSI_DN3			
B 7	MIPIDSI_DPCLK	S	MIPIDSI_DPCLK			
A 7	MIPIDSI_DNCLK	S	MIPIDSI_DNCLK			
C 7	MIPIDSI_VREG_0P4V	S	MIPIDSI_VREG_0P4V			

2.3.2.7 MIPI CSI

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
B 2	MIPICSI_DP0	S	MIPICSI_DP0			
A 2	MIPICSI_DN0	S	MIPICSI_DN0			
B 3	MIPICSI_DP1	S	MIPICSI_DP1			
A 3	MIPICSI_DN1	S	MIPICSI_DN1			
B 4	MIPICSI_DP2	S	MIPICSI_DP2			
A 4	MIPICSI_DN2	S	MIPICSI_DN2			
B 5	MIPICSI_DP3	S	MIPICSI_DP3			
A 5	MIPICSI_DN3	S	MIPICSI_DN3			
B 1	MIPICSI_DPCLK	S	MIPICSI_DPCLK			
A 1	MIPICSI_DNCLK	S	MIPICSI_DNCLK			

2.3.2.8 VIP

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AA 9	VID0_0	S	GPIOD28	VID0_0	TSIDATA1_0	SA24
AC 9	VID0_1	S	GPIOD29	VID0_1	TSIDATA1_1	
AD 9	VID0_2	S	GPIOD30	VID0_2	TSIDATA1_2	
AC 11	VID0_3	S	GPIOD31	VID0_3	TSIDATA1_3	
AB 9	VID0_4	S	GPIOE0	VID0_4	TSIDATA1_4	
AE 9	VID0_5	S	GPIOE1	VID0_5	TSIDATA1_5	
AE 10	VID0_6	S	GPIOE2	VID0_6	TSIDATA1_6	
AE 11	VID0_7	S	GPIOE3	VID0_7	TSIDATA1_7	
AD 10	VICLK0	S	GPIOE4	VICLK0	TSICLK1	
AA 11	VIHSYNC0	S	GPIOE5	VIHSYNC0	TSISYNC1	
AD 11	VIVSYNC0	S	GPIOE6	VIVSYNC0	TSIDP1	
J21	VID1_0	S	GPIOA30	VID1_0	SDEX0	I2SBCLK1

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
L20	VID1_1	S	GPIOB0	VID1_1	SDEX1	I2SLRCLK1
M20	VID1_2	S	GPIOB2	VID1_2	SDEX2	I2SCLK2
P20	VID1_3	S	GPIOB4	VID1_3	SDEX3	I2SLRCLK2
R20	VID1_4	S	GPIOB6	VID1_4	SDEX4	I2SDOUT1
U20	VID1_5	S	GPIOB8	VID1_5	SDEX5	I2SDOUT2
V20	VID1_6	S	GPIOB9	VID1_6	SDEX6	I2SDIN1
W21	VID1_7	S	GPIOB10	VID1_7	SDEX7	I2SDIN2
J20	VICLK1	S	GPIOA28	VICLK1	I2SMCLK2	I2SMCLK1
E 14	VIHsync1	S	GPIOE13	GMAC_COL	VIHsync1	
D 11	VIVsync1	S	GPIOE7	GMAC_TXD0	VIVsync1	
AA 14	VID2_0	S	SA17	GPIOC17	TSIDP0	VID2_0
AB 12	VID2_1	S	SA18	GPIOC18	SDCLK2	VID2_1
AB 14	VID2_2	S	SA19	GPIOC19	SDCMD2	VID2_2
AB 11	VID2_3	S	SA20	GPIOC20	SDDAT2_0	VID2_3
Y 11	VID2_4	S	SA21	GPIOC21	SDDAT2_1	VID2_4
AC 14	VID2_5	S	SA22	GPIOC22	SDDAT2_2	VID2_5
AC 12	VID2_6	S	SA23	GPIOC23	SDDAT2_3	VID2_6
AE 12	VID2_7	S	LATADDR	GPIOC24	SPDIFRX	VID2_7
AD 12	VICLK2	S	SA14	GPIOC14	PWM2	VICLK2
AD 14	VIHsync2	S	SA15	GPIOC15	TSICLK0	VIHsync2
AE 14	VIVsync2	S	SA16	GPIOC16	TSISYNC0	VIVsync2

2.3.2.9 Ethernet MAC

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
D 11	GMAC_TXD0	S	GPIOE7	GMAC_TXD0	VIVsync1	
C 11	GMAC_TXD1	S	GPIOE8	GMAC_TXD1		
C 12	GMAC_TXD2	S	GPIOE9	GMAC_TXD2		
D 12	GMAC_TXD3	S	GPIOE10	GMAC_TXD3		
E 12	GMAC_TXEN	S	GPIOE11	GMAC_TXEN		
E 11	GMAC_TXER	S	GPIOE12	GMAC_TXER		
E 14	GMAC_COL	S	GPIOE13	GMAC_COL	VIHsync1	
D 17	GMAC_RXD0	S	GPIOE14	GMAC_RXD0	SPICLK1	
C 17	GMAC_RXD1	S	GPIOE15	GMAC_RXD1	SPIFRM1	
D 18	GMAC_RXD2	S	GPIOE16	GMAC_RXD2		
C 18	GMAC_RXD3	S	GPIOE17	GMAC_RXD3		
E 18	GMAC_RXCLK	S	GPIOE18	GMAC_RXCLK	SPIRXD1	

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
E 17	GMAC_RXDV	S	GPIOE19	GMAC_RXDV	SPITXD1	
D 15	GMAC_MDC	S	GPIOE20	GMAC_MDC		
D 14	GMAC_MDIO	S	GPIOE21	GMAC_MDIO		
E 15	GMAC_RXER	S	GPIOE22	GMAC_RXER		
B 12	GMAC_CRS	S	GPIOE23	GMAC_CRS		
A 12	GMAC_GTXCLK	S	GPIOE24	GMAC_GTXCLK		

2.3.2.10 MPEG-TS Interface

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AA 25	TSIDATA0_0	S	SD8	GPIOB24	TSIDATA0_0	
AB 25	TSIDATA0_1	S	SD9	GPIOB25	TSIDATA0_1	
AC 25	TSIDATA0_2	S	SD10	GPIOB26	TSIDATA0_2	
AD 25	TSIDATA0_3	S	SD11	GPIOB27	TSIDATA0_3	
AE 25	TSIDATA0_4	S	SD12	GPIOB28	TSIDATA0_4	UARTRXD4
AE 24	TSIDATA0_5	S	SD13	GPIOB29	TSIDATA0_5	UARTTXD4
AD 24	TSIDATA0_6	S	SD14	GPIOB30	TSIDATA0_6	UARTRXD5
AC 24	TSIDATA0_7	S	SD15	GPIOB31	TSIDATA0_7	UARTTXD5
AD 14	TSICLK0	S	SA15	GPIOC15	TSICLK0	VIHsync2
AE 14	TSISYNC0	S	SA16	GPIOC16	TSISYNC0	VIVSync2
AA 14	TSIDP0	S	SA17	GPIOC17	TSIDP0	VID2_0
AA 9	TSIDATA1_0	S	GPIOD28	VID0_0	TSIDATA1_0	SA24
AC 9	TSIDATA1_1	S	GPIOD29	VID0_1	TSIDATA1_1	
AD 9	TSIDATA1_2	S	GPIOD30	VID0_2	TSIDATA1_2	
AC 11	TSIDATA1_3	S	GPIOD31	VID0_3	TSIDATA1_3	
AB 9	TSIDATA1_4	S	GPIOE0	VID0_4	TSIDATA1_4	
AE 9	TSIDATA1_5	S	GPIOE1	VID0_5	TSIDATA1_5	
AE 10	TSIDATA1_6	S	GPIOE2	VID0_6	TSIDATA1_6	
AE 11	TSIDATA1_7	S	GPIOE3	VID0_7	TSIDATA1_7	
AD 10	TSICLK1	S	GPIOE4	VICLK0	TSICLK1	
AA 11	TSISYNC1	S	GPIOE5	VIHsync0	TSISYNC1	
AD 11	TSIDP1	S	GPIOE6	VIVSync0	TSIDP1	

2.3.2.11 UART_ISO7816

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AD 19	UARTTXD0	S	GPIOD18	UARTTXD0	ISO7816	SDWP2

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AE 19	UARTRXD0	S	GPIOD14	UARTRXD0	ISO7816	
AD 18	UARTTXD1	S	GPIOD19	UARTTXD1	ISO7816	SDnCD2
AE 18	UARTRXD1	S	GPIOD15	UARTRXD1	ISO7816	
AC 22	UARTnDCD1	S	SA4	GPIOC4	UARTnDCD1	SDnINT0
AD 22	UARTnCTS1	S	SA5	GPIOC5	UARTnCTS1	SDWP0
AE 22	UARTnRTS1	S	SA6	GPIOC6	UARTnRTS1	SDnCD0
AE 21	UARTnDSR1	S	SA7	GPIOC7	UARTnDSR1	SDnRST1
AD 21	UARTnDTR1	S	SA8	GPIOC8	UARTnDTR1	SDnINT1
AB 21	UARTnRI1	S	GPIOC28	NSCS1	UARTnRI1	
Y 18	UARTTXD2	S	GPIOD20	UARTTXD2	CANTX0	SDWP1
Y 19	UARTRXD2	S	GPIOD16	UARTRXD2	CANRX0	
W 17	UARTTXD3	S	GPIOD21	UARTTXD3	CANTX1	SDnCD1
Y 17	UARTRXD3	S	GPIOD17	UARTRXD3	CANRX1	
AE 24	UARTTXD4	S	SD13	GPIOB29	TSIDATA0_5	UARTTXD4
AE 25	UARTRXD4	S	SD12	GPIOB28	TSIDATA0_4	UARTRXD4
AC 24	UARTTXD5	S	SD15	GPIOB31	TSIDATA0_7	UARTTXD5
AD 24	UARTRXD5	S	SD14	GPIOB30	TSIDATA0_6	UARTRXD5

2.3.2.12 I2C

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AC 19	SDA0	S	GPIOD3	SDA0	ISO7816	
AC 20	SCL0	S	GPIOD2	SCL0	ISO7816	
AB 18	SDA1	S	GPIOD5	SDA1		
AB 17	SCL1	S	GPIOD4	SCL1		
AB 19	SDA2	S	GPIOD7	SDA2		
AC 18	SCL2	S	GPIOD6	SCL2		

2.3.2.13 SPI/SSP

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AE 16	SPITXD0	S	GPIOC31	SPITXD0		
AD 16	SPIRXD0	S	GPIOD0	SPIRXD0	PWM3	
AE 17	SPICLK0	S	GPIOC29	SPICLK0		
AD 17	SPIFRM0	S	GPIOC30	SPIFRM0		
E 17	SPITXD1	S	GPIOE19	GMAC_RXDV	SPITXD1	
E 18	SPIRXD1	S	GPIOE18	GMAC_RXCLK	SPIRXD1	

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
D 17	SPICLK1	S	GPIOE14	GMAC_RXD0	SPICLK1	
C 17	SPIFRM1	S	GPIOE15	GMAC_RXD1	SPIFRM1	
V 15	SPITXD2	S	SA12	GPIOC12	SPITXD2	SDnRST2
W 14	SPIRXD2	S	SA11	GPIOC11	SPIRXD2	USB2.0OTG_DnVBUS
AB 24	SPICLK2	S	SA9	GPIOC9	SPICLK2	PDMStrobe
V 19	SPIFRM2	S	SA10	GPIOC10	SPIFRM2	

2.3.2.14 PWM

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AE 15	PWM0	S	GPIOD1	PWM0	SA25	
V 14	PWM1	S	SA13	GPIOC13	PWM1	SDnINT2
AD 12	PWM2	S	SA14	GPIOC14	PWM2	VICLK2
AD 16	PWM3	S	GPIOD0	SPIRXD0	PWM3	

2.3.2.15 PPM

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
Y 12	PPM	S	GPIOD8	PPM		

2.3.2.16 PDM

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AB 23	PDMDATA0	S	RDNWR	GPIOC26	PDMDATA0	
AA 24	PDMDATA1	S	NSDQM	GPIOC27	PDMDATA1	
AB 24	PDMStrobe	S	SA9	GPIOC9	SPICLK2	PDMStrobe

2.3.2.17 SPDIF

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AC 21	SPDIFTX	S	NSWAIT	GPIOC25	SPDIFTX	
AE 12	SPDIFRX	S	LATADDR	GPIOC24	SPDIFRX	VID2_7

2.3.2.18 SD/MMC

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
T25	SDDATO_0	S	GPIOB1	SDDATO_0		

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
U24	SDDAT0_1	S	GPIOB3	SDDAT0_1		
U25	SDDAT0_2	S	GPIOB5	SDDAT0_2		
V24	SDDAT0_3	S	GPIOB7	SDDAT0_3		
T24	SDCLK0	S	GPIOA29	SDCLK0		
U23	SDCMD0	S	GPIOA31	SDCMD0		
AD 22	SDWP0	S	SA5	GPIOC5	UARTnCTS1	SDWP0
AE 22	SDnCD0	S	SA6	GPIOC6	UARTnRTS1	SDnCD0
Y21	SDnRST0	S	SA3	GPIOC3	HDMI_CEC	SDnRST0
AC 22	SDnINT0	S	SA4	GPIOC4	UARTnDCD1	SDnINT0
AA 18	SDDAT1_0	S	GPIOD24	SDDAT1_0		
AA 17	SDDAT1_1	S	GPIOD25	SDDAT1_1		
Y15	SDDAT1_2	S	GPIOD26	SDDAT1_2		
Y14	SDDAT1_3	S	GPIOD27	SDDAT1_3		
AA 20	SDCLK1	S	GPIOD22	SDCLK1		
AA 19	SDCMD1	S	GPIOD23	SDCMD1		
Y 18	SDWP1	S	GPIOD20	UARTTXD2	CANTX0	SDWP1
W 17	SDnCD1	S	GPIOD21	UARTTXD3	CANTX1	SDnCD1
AE 21	SDnRST1	S	SA7	GPIOC7	UARTnDSR1	SDnRST1
AD 21	SDnINT1	S	SA8	GPIOC8	UARTnDTR1	SDnINT1
AB 11	SDDAT2_0	S	SA20	GPIOC20	SDDAT2_0	VID2_3
Y 11	SDDAT2_1	S	SA21	GPIOC21	SDDAT2_1	VID2_4
AC 14	SDDAT2_2	S	SA22	GPIOC22	SDDAT2_2	VID2_5
AC 12	SDDAT2_3	S	SA23	GPIOC23	SDDAT2_3	VID2_6
AB 12	SDCLK2	S	SA18	GPIOC18	SDCLK2	VID2_1
AB 14	SDCMD2	S	SA19	GPIOC19	SDCMD2	VID2_2
AD 19	SDWP2	S	GPIOD18	UARTTXD0	ISO7816	SDWP2
AD 18	SDnCD2	S	GPIOD19	UARTTXD1	ISO7816	SDnCD2
V 15	SDnRST2	S	SA12	GPIOC12	SPITXD2	SDnRST2
V 14	SDnINT2	S	SA13	GPIOC13	PWM1	SDnINT2

2.3.2.19 USB2.0 HOST

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
F25	USB2.0HOST_DP	S	USB2.0HOST_DP			
F24	USB2.0HOST_DM	S	USB2.0HOST_DM			
G25	USB2.0HOST_RKELVIN	S	USB2.0HOST_RKELVIN			

2.3.2.20 USB2.0 HSIC HOST

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
H24	USBHSIC_DATA	S	USBHSIC_DATA			
H25	USBHSIC_STROBE	S	USBHSIC_STROBE			

2.3.2.21 USB2.0 OTG

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
D25	USB2.0OTG_DP	S	USB2.0OTG_DP			
D24	USB2.0OTG_DM	S	USB2.0OTG_DM			
E25	USB2.0OTG_RKELVIN	S	USB2.0OTG_RKELVIN			
C25	USB2.0OTG_ID	S	USB2.0OTG_ID			
C24	USB2.0OTG_VBUS	S	USB2.0OTG_VBUS			
W14	USB2.0OTG_DrvVBUS	S	SA11	GPIOC11	SPIRXD2	USB2.0OTG_DrvVBUS
E24	USB2.0OTG_USBVBUS	S	USB2.0OTG_USBVBUS			

2.3.2.22 I2S & AC97

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AD15	I2SDOUT0	S	GPIOD9	I2SDOUT0	AC97_DOUT	
AC15	I2SDIN0	S	GPIOD11	I2SDIN0	AC97_DIN	
AB15	I2SBCLK0	S	GPIOD10	I2SBCLK0	AC97_BCLK	
AA15	I2SMCLK0	S	GPIOD13	I2SMCLK0	AC97_nRST	
AC17	I2SLRCLK0	S	GPIOD12	I2SLRCLK0	AC97_SYNC	
R20	I2SDOUT1	S	GPIOB6	VID1_4	SDEX4	I2SDOUT1
V20	I2SDIN1	S	GPIOB9	VID1_6	SDEX6	I2SDIN1
J21	I2SBCLK1	S	GPIOA30	VID1_0	SDEX0	I2SBCLK1
J20	I2SMCLK1	S	GPIOA28	VIDCLK1	I2SMCLK2	I2SMCLK1
L20	I2SLRCLK1	S	GPIOB0	VID1_1	SDEX1	I2SLRCLK1
U20	I2SDOUT2	S	GPIOB8	VID1_5	SDEX5	I2SDOUT2
W21	I2SDIN2	S	GPIOB10	VID1_7	SDEX7	I2SDIN2
M20	I2SBCLK2	S	GPIOB2	VID1_2	SDEX2	I2SBCLK2
J20	I2SMCLK2	S	GPIOA28	VIDCLK1	I2SMCLK2	I2SMCLK1
P20	I2SLRCLK2	S	GPIOB4	VID1_3	SDEX3	I2SLRCLK2

2.3.2.23 ADC

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AE 3	ADC0	S	ADC0			
AB 5	ADC1	S	ADC1			
AD 3	ADC2	S	ADC2			
AE 2	ADC3	S	ADC3			
AC1	ADC4	S	ADC4			
AD1	ADC5	S	ADC5			
AD 2	ADC6	S	ADC6			
AD 6	ADC7	S	ADC7			

2.3.2.24 ALIVE GPIO

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AB 8	ALIVEGPIO0	S	ALIVEGPIO0			
AA 8	ALIVEGPIO1	S	ALIVEGPIO1			
AA 7	ALIVEGPIO2	S	ALIVEGPIO2			
Y 9	ALIVEGPIO3	S	ALIVEGPIO3			
Y 8	ALIVEGPIO4	S	ALIVEGPIO4			
W 9	ALIVEGPIO5	S	ALIVEGPIO5			

2.3.2.25 JTAG

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
V 6	NTRST	S	NTRST	GPIOE25		
Y 5	TMS	S	TMS	GPIOE26		
W 5	TDI	S	TDI	GPIOE27		
AA 6	TCLK	S	TCLK	GPIOE28		
Y 7	TDO	S	TDO	GPIOE29		

2.3.2.26 Crystal PLL & RTC

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
A20	PLLXTI	S	PLLXTI			
B20	PLLXTO	S	PLLXTO			
AE 4	RTCXTI	S	RTCXTI			
AD 4	RTCXTO	S	RTCXTO			

2.3.2.27 Miscellaneous

Ball	Name	Type	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3
AE1	NRESET	S	NRESET			
AB7	NGRESETOUT	S	NGRESETOUT			
AB6	VDDPWRON	S	VDDPWRON			
AC6	VDDPWRON_DDR	S	VDDPWRON_DDR			
AE6	NVDDPWRTOGGLE	S	NVDDPWRTOGGLE			
AB2	NBATF	S	NBATF			
AE7	TEST_EN	S	TEST_EN			
AD8	WIRE0	S	WIRE0			
AE8	WIRE1	S	WIRE1			
AD7	EFUSE_FSOURCE	S	EFUSE_FSOURCE			

2.3.2.28 Power : VDD

Ball	Name	Type	Description
D5, D6, E6, E7, E8, G11, J12, J14, K11, K13, K15, L12, L14, L16, L17, M11, M13, M15, N12, P11, R12, T11, U12, V11, V12, W11, W12	VDDI	P	1.0V for CORE
N14, N16, P13, P15, P16, R14, R15, R16, T14, T15, T16, U14, U15	VDDI_ARM	P	1.0V ~ 1.3V for CPU.
K9, K10, L9, L10, M9, M10, N9, N10, P9, P10, R9, R10, T9, T10	VDDQ	P	1.5V for DDR3 IO
U10, U11, V9	VDDP18	P	1.8V for Internal IO
L19, M19, P19, R18, R19, U18, U19, V17	DVDD33_IO	P	3.3V for IO
E23	DVDD10_USB0	P	1.0V for USB
G23	DVDD10_USBHOST0	P	1.0V for USB HOST
F23	VDD18_USB0	P	1.8V for USB
F21	VDD18_USBHOST	P	1.8V for USB HOST
E22	VDD33_USB0	P	3.3V for USB
G24	VDD33_USBHOST	P	3.3V for USB HOST
H23	DVDD12_HSIC	P	1.2V for USB HSIC HOST
AC8	VDDI10_ALIVE	P	1.0V for ALIVE
V7	VDDP18_ALIVE	P	1.8V for Internal IO ALIVE
W8	VDD33_ALIVE	P	3.3V for ALIVE
AC4, AC7	VDD18_RTC	P	1.8V for RTC
D20	VDD18_OSC	P	1.8V for Crystal
G14	AVDD10_LV	P	1.0V for LVDS
F15	AVDD18_LV	P	1.8V for LVDS
C22	AVDD10_HM	P	1.0V for HDMI

Ball	Name	Type	Description
F19	VDD10_HM_PLL	P	1.0V for HDMI PLL
D23	VDD18_HM	P	1.8V for HDMI
C8	M_VDD10_PLL	P	1.0V for MIPI PLL
C4, C5, D7, D8	M_VDD10	P	1.0V for MIPI
C6	M_VDD18	P	1.8V for MIPI
AC5	AVDD18_ADC	P	1.8V for ADC
AC2	ADCREF	P	1.8V for ADC reference VDD
C21, D19, D21, E20	AVDD18_PLL	P	1.8V for PLL
T13, U13	DVDD_VID2_SD2	P	2.8V for VID2/SD2
AA12	DVDD_VID0	P	2.8V for VID0
F14	DVDD_GMAC	P	2.8V for Ethernet MAC

2.3.2.29 Power : GND

Ball	Name	Type	Description
A6, A19, B6, B19, C9, D9, E9, F7, F8, F9, F22, G8, G9, G12, G17, G21, G22, H6, H7, H9, H11, H12, H14, H15, H17, H19, H20, J7, J8, J10, J11, J13, J15, J16, J18, J19, K12, K14, K16, K17, L6, L7, L8, L11, L13, L15, L18, M6, M7, M8, M12, 14, M16, M17, M18, N11, N13, N15, N17, P6, P7, P8, P12, P14, P17, P18, R6, R7, R8, R11, R13, R17, T12, T17, U6, U7, U8, U16, W15, W18	VSSI	G	Digital GND
C20	VSS18_OSC	G	GND for 1.8V Crystal VDD
G15	AVSS10_LV	G	GND for 1.0V LVDS VDD
F11, F12	AVSS18_LV	G	GND for 1.8V LVDS VDD
AD5	AVSS18_ADC	G	GND for 1.8V ADC VDD
E19, F17, F18, G18	AVSS18_PLL	G	GND for 1.8V PLL VDD

Section 3. System Boot

3.1 Overview

NXP4330D/Q supports various system boot modes. Boot Mode is determined by System Configuration when boot reset off.

- External Static Memory Boot
- Internal ROM Boot
 - NAND boot with Error Correction
 - SD/SDHC/MMC boot
 - moviNAND/iNAND boot
 - SPI Serial EEPROM boot
 - UART boot
 - USB boot

3.2 Functional Description

3.2.1 System Configuration

Pins	RST_CFG	Static Memory	UART	Serial Flash	SD MMC	USB Device	Nand on SD	Nand on SDEX
SD1, SD0	RST_CFG[1:0]	-	-	-	-	-	NANDTYPE [1:0]	-
SD2	RST_CFG2	-	-	-	-	-	NANDPAGE	-
SD3	RST_CFG3	-	-	-	-	-	SELCS	-
SD4	RST_CFG4	-	DECRYPT	DECRYPT	DECRYPT	DECRYPT	DECRYPT	-
SD5	RST_CFG5	-	I-Cache	I-Cache	I-Cache	I-Cache	I-Cache	-
SD6	RST_CFG6	-	-	-	-	OTG Session Check	-	-
SD7	RST_CFG7	-	-	-	eMMC BootMode	-	-	-
DISD0	RST_CFG8	LATADDR	-	-	-	-	-	-
DISD1	RST_CFG9	-	-	ADDRWIDTH 0	PARTITION	-	-	-
DISD2	RST_CFG10	-	BAUDRATE	ADDRWIDTH 1	eMMCBOOT	-	-	-
DISD3	RST_CFG11	BUSWIDTH	-	-	-	-	-	-
DISD7 DISD6 DISD5 DISD4	RST_CFG[15:12]	0	3	4	5	6	7	15
VID1[1] VID1[0]	RST_CFG[17:16]	-	-	-	-	-	-	NANDTYPE [1:0]
VID1[2]	RST_CFG18	-	-	-	-	-	-	NANDPAGE
VID1[3]	RST_CFG19	-	-	-	-	-	-	-
VID1[4]	RST_CFG20	-	-	-	-	-	-	DECRYPT
VID1[5]	RST_CFG21	-	-	-	-	-	-	I-Cache
VID1[6]	RST_CFG22	-	-	-	-	-	-	-
VID1[7]	RST_CFG23	-	-	-	-	-	-	-

Table 3-1 System Configuration by RST_CFG Pins

Name	Pin	Note	
NANDTYPE[1:0]	RST_CFG[1:0] /RST_CFG[17:16]	NAND flash Memory Type	0: Small Block 3 Address 1: Small block 4 Address 2: Large 4 Address 3: Large 5 Address
NANDPAGE	RST_CFG2 /RST_CFG18	Pagesize of Large NAND Flash	0: 2K 1: 4K or above

Name	Pin	Note	
SELCS	RST_CFG3	NAND Chip Select	When SD bus -> 0:nNCS0 1:nNCS1. When SDEX bus ->nNCS1.
DECRYPT	RST_CFG4 /RST_CFG20- only checked sdex	AES ECB mode decrypt	0: not decrypt 1: decrypt
I-CACHE	RST_CFG5 /RST-CFG21- only checked sdex	I-Cache Enable	0: Disable 1: Enable
OTG SESSION CHECK	RST_CFG6	OTG Session check	0: not check 1: check
EMMC BOOT MODE	RST_CFG7	eMMC Boot mode	0: Normal SD Boot 1: eMMC Boot
LATADDR	RST_CFG8	Static Latched Address	0: None 1: Latched
PARTITION	RST_CFG9	Boot Partition on eMMC	0: Default Partition 1: Boot Partition (Partition#1)
EMMCBOOT	RST_CFG10	eMMC Boot mode	0: Alternative Boot 1: Normal Boot
BAUD	RST_CFG10	UART Baudrate	0: 19200bps 1: 115200bps
ADDRWIDTH[1:0]	RST_CFG[10:9]	Serial Flash Address width	0: 16bit 1: 24bit 2: 32bit
BOOTMODE[3:0]	RST_CFG[15:12]	Boot Mode Select	0: Static Memory 3: UART 4: SPI 5: SDMMC 6: USB 7: NAND (SD bus) 15: NAND (SDEX bus)

Table 3-2 System Configuration by Function

3.3 External Static Memory Boot

CPU executes External Static Memory Access without CPU Hold.

3.3.1 External Static Memory Boot Features

Supports 16/8 bit Static Memory

3.3.2 External Static Memory Boot System configuration

Pin Name	Function Name	Description
RST_CFG[7:0]	-	Don't care
RST_CFG8	CfgSTLATADD	Static Latched Address(user select) 0: None 1: Latched
RST_CFG[10:9]	-	Don't care
RST_CFG11	CfgSTBUSWidth	Static Bus Width(user select) 0: 8bit 1: 16bit
RST_CFG[14:12]	BOOTMODE[2:0]	Pull-down
RST_CFG[24:16]	-	Don't care

Table 3-3. External static memory not system configuration setting description

3.3.3 External Static Memory Boot Operation

In case of External Static Memory Boot, nSCS[0] is set to Address 0x00000000 by reset configuration and CPU can access Static Memory through MCU-S.

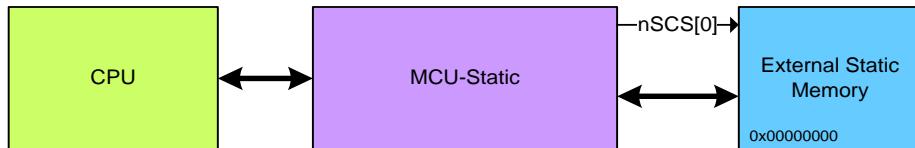


Figure 3-1. External Static Memory Boot

3.4 Internal ROM Boot

The chip has built-in 20KB ROM. It is possible to set Internal ROM address to 0th address by setting CfgBOOTMODE of System Configuration to '3~7.' After Reset, CPU executes instruction fetched from 0th address of the Internal ROM. Internal ROM has a code supporting various Booting methods. This ROM code executes User Bootcode by reading it through various media and loading it to specific memory. This Booting method is defined as internal ROM Booting (which is, from now on, called iROMBOOT).

iROMBOOT uses internal SRAM for storing stack or data. Therefore it is possible for the content of internal SRAM to change after iROMBOOT is executed.

3.4.1 Features

- Supports five booting modes: SPI Serial EEPROM BOOT, UART BOOT, USBBOOT, SDHCBOOT and NANDBOOT with Error Correction
- Supports CPU Exception Vector Redirection for OS systems without using MMU.
- Supports Fast Power Control: Set VDDPWRON and VDDPWRON_DDR as High.

3.4.2 System configuration for the internal ROM booting

iROMBOOT supports 5 Booting modes such as USBBOOT, UART BOOT, SPI Serial EEPROM BOOT, SDHCBOOT, and NANDBOOT with Error Correction. Every Booting mode supports various booting methods by referring to Reset states from SD[15:0] and SDEX[7: 0]. Table 3-5 shows System configuration for each Booting mode.

Pins	iROMBOOT				
	UART	SPI Serial Flash	SDMMC	USB Device	NANDBOOT with Error Correction
RST_CFG[1:0]/ RST_CFG[17:16]	Don't care				NANDTYPE[1:0]
RST_CFG[2, 18]					PAGESIZE
RST_CFG[3]					SELCS
RST_CFG[4, 20]	DECRYPT				
RST_CFG[5, 21]	ICACHE				
RST_CFG[6]	Don't care		OTG Session Check		Don't care
RST_CFG[7]	Don't care		eMMC Boot Mode		
RST_CFG[8]	LATADDR				
RST_CFG[9]	Don't care	ADDRWIDTH0	PARTITION	Don't care	
RST_CFG[10]		ADDRWIDTH1	eMMCBOOT		
RST_CFG[11]	BUSWIDTH = 0				
RST_CFG[14:12]	BOOTMODE=3	BOOTMODE=4	BOOTMODE=5	BOOTMODE=6	BOOTMODE=7
RST_CFG[15]	Don't care				SEL SDEX

Table 3-4. iROMBOOT system configuration

3.4.3 USB BOOT

iROMBOOT can load User Bootcode via USB to memory and execute this code, which Booting method is called USBBOOT.

3.4.3.1 Features

- Supports Full speed or High Speed USB connection.
- Uses the USB Bulk transfer.
- Supports 64 bytes for Full speed and 512 bytes for High speed as Max packet size.

3.4.3.2 Operation

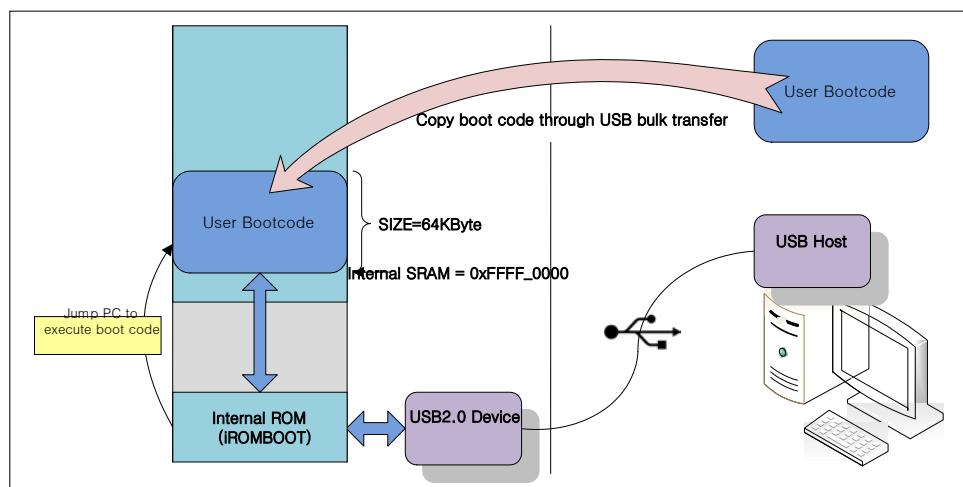


Figure 3-2. USBBOOT Operation

USB Host Program should transfer User boot code by using bulk transfer through EP2 of USB Device. The max packet size is changeable according to USB connection speed at Endpoint. In Full speed connection, USB Host Program can transfer the maximum 64 bytes as one packet and, in High speed connection, the maximum 512 bytes as one packet. USB Host Program should transfer the data packet of even size even though it can transfer the same packet as the max size or the packet smaller than the max size.

USBBOOT writes User Bootcode from USB Host Program and USBBOOT executes User Bootcode by changing PC to 0xFFFF0000 after it receives User Bootcode size of 16KB.

3.4.3.3 USB Descriptors

USB Host Program can get Descriptor of USBBOOT by using Get_Descriptor Request. Table 3-6 shows Descriptor of USBBOOT. USBBOOT has one configuration, one interface, and two additional Endpoints except Control Endpoint. However, Endpoint 1 exists only for compatibility. Then USBBOOT only receives data by using Endpoint2 only.

Offset	Field	Size	USBBOOT Value		Description
			Full Speed	High Speed	
Device Descriptor					
0	bLength	1	18		Size of this descriptor in bytes
1	bDescriptorType	1	01h		DEVICE descriptor type
2	bcdUSB	2	0110h	0200h	USB spec release number in BCD
4	bDeviceClass	1	FFh		Class code

5	bDeviceSubClass	1	FFh	Subclass code	
6	bDeviceProtocol	1	FFh	Protocol code	
7	bMaxPacketSize0	1	64	Maximum packet size for EP0	
8	idVendor	2	2375h	Vender ID	
10	idProduct	2	4330h	Product ID	
12	bcdDevice	2	0000h	Device release number in BCD	
14	iManufacturer	1	0	Index of string descriptor describing manufacturer	
15	iProduct	1	0	Index of string descriptor describing product	
16	iSerialNumber	1	0	Index of string descriptor describing the device's serial number	
17	bNumConfiguration	1	1	Number of possible configuration	
Configuration Descriptor					
0	bLength	1	9	Size of this descriptor in bytes	
1	bDescriptorType	1	02h	CONFIGURATION descriptor type	
2	wTotalLength	2	32	Total length of data returned for this configuration	
4	bNumInterfaces	1	1	Number of interfaces	
5	bConfigurationValue	1	1	Value to use as an argument to the Set Configuration	
6	iConfiguration	1	0	Index of string descriptor describing this configuration	
7	bmAttribute	1	80h	Configuration characteristics	
8	bMaxPower	1	25	Maximum power consumption	
Interface Descriptor					
0	bLength	1	9	Size of this descriptor in bytes	
1	bDescriptorType	1	04h	INTERFACE descriptor type	
2	blnterfaceNumber	1	0	Number of this interface	
3	bAlternateSetting	1	0	Value used to select this alternate setting	
4	bNumEndpoints	1	2	Value used to select this alternate setting for the interface	
5	blnterfaceClass	1	FFh	Class code	
6	blnterfaceSubClass	1	FFh	Subclass code	
7	blnterfaceProtocol	1	FFh	Protocol code	
8	ilnterface	1	0	Index of string descriptor describing this interface	
Endpoint Descriptor for EP1					
0	bLength	1	7	Size of this descriptor in bytes	
1	bDescriptorType	1	05h	ENDPOINT descriptor type	
2	bEndpointAddress	1	81h	The address of the endpoint	
3	bmAttributes	1	02h	the endpoint's attributes	
4	wMaxPacketSize	2	64	512	Maximum packet size
6	blnterval	1	0	Interval for polling endpoint for data transfers	
Endpoint Descriptor for EP2					
0	bLength	1	7	Size of this descriptor in bytes	
1	bDescriptorType	1	05h	ENDPOINT descriptor type	
2	bEndpointAddress	1	02h	The address of the endpoint	
3	bmAttributes	1	02h	the endpoint's attributes	
4	wMaxPacketSize	2	64	512	Maximum packet size
6	blnterval	1	0	Interval for polling endpoint for data transfers	

Table 3-5. USBBOOT description

3.4.4 SDHCBOOT

iROMBOOT can execute User Bootcode by reading it from SD memory card, MMC memory card, moviNAND, and iNAND and loading it to memory by using SDHC module. This method is called SDHCBOOT.

3.4.4.1 Features

- Supports SD/MMC memory card, eSD (iNAND), and eMMC (moviNAND)
- Supports High Capacity SD/MMC memory card

- Supports eSD Booting
 - Supports FAST_BOOT
 - Supports Boot Partition
- Supports eMMC Booting
- Supports eMMC Booting
 - Supports Normal Booting and Alternate Booting
 - Supports only 4-bit data bus
 - Doesn't support BOOT_ACK
- Outputs 400 KHz SDCLK for Identification and 22.9 MHz SDCLK for Data Transfer

3.4.4.2 Operation

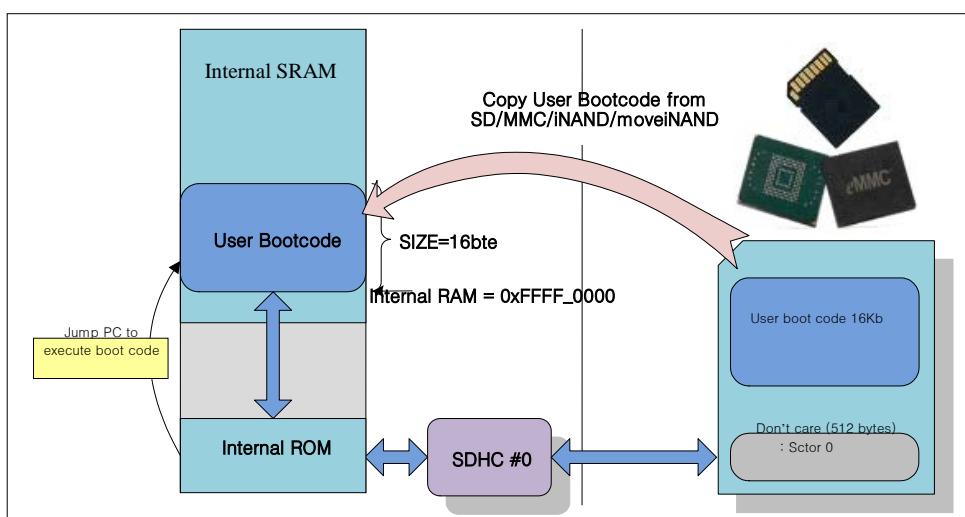


Figure 3-3. SDHCBOOT Operation

SDHCBOOT uses SDHC #0 module. The pins SDHC #0 module uses are GPIO A[29, 31] and GPIO B[1, 3, 5, 7].

SDHCBOOT provides various Booting methods according to CFG pins, in which the specification of each method is recommended to refer to Table 3-1.

User Bootcode should be written as Table 3-7 to Storage Device for the use of SDHCBOOT.

Sector	Name	Description
0	RESERVED	SDHCBOOT don't care data in 0 th Sector. Therefore it is possible to use 0 th Sector for storing MBR (Master Boot record), and to include User Bootcode along with File System into one Physical Partition.
1 ~ 32	User Bootcode	Boot code User made has 16KB size from 2 nd Sector to 32st Sector

Table 3-6. Boot Data Format for SDHCBOOT

The SDHCBOOT Booting process is as follows.

- eMMC Booting : SDHCBOOT executes eMMC Booting in case that CfgSDHCBM is 1.
 - When CfgEMMCBM is '1', Normal eMMC Booting is executed, and when CfgEMMCBM is '0', Alternate eMMC Booting is executed.
 - For eMMC Booting, SDHCBOOT always uses 4-bit data bus. Therefore BOOT_BUS_WIDTH of EXT_CSD should be set to '1.' And BOOT_ACK of EXT_CSD should be set to '0' because BOOT_ACK is not available for eMMCBoot.

- Normal SDMMC Booting is processed when no Data is transferred from Card in 1 second.
- 512 bytes first transferred from Card are not used.
- 512 bytes secondly transferred from Card are used.
- User Bootcode transferred from Card is loaded to inter sram to be executed.

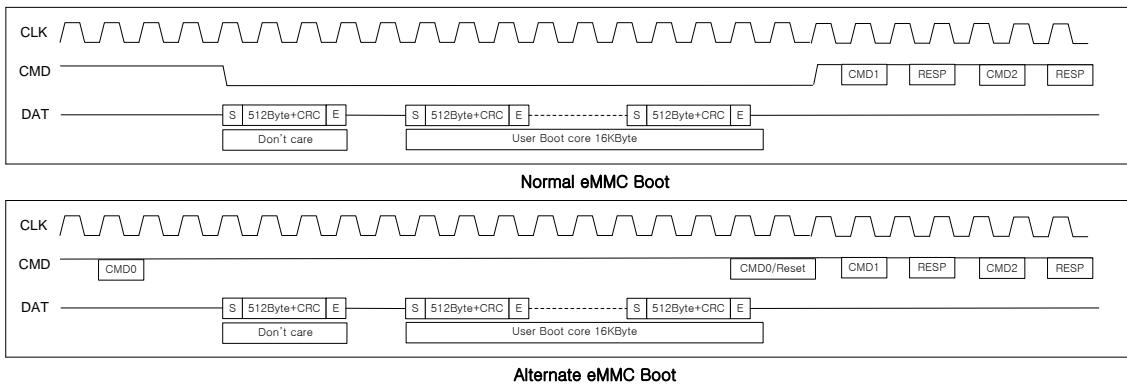


Figure 3-4. eMMC Boot

- Normal SDMMC Booting is executed when CfgSDHCBM is '0', or eMMC Booting fails.
 - Go idle state
 - SDHCBOOT identifies the type of Card and initializes.
 - The state of Card changes to Data Transfer Mode.
 - SDHCBOOT selects partition according to CfgPARTITION.
 - SDHCBOOT reads User Bootcode from Sector #1, and load it to internal sram to be executed.

3.4.5 NANDBOOT with Error Correction

iROMBOOT provides the booting method which can correct any error in User bootcode stored in NAND Flash memory. This Booting method is described as NANDBOOT with Error Correction (which is abbreviated to NANDBOOTEC).

3.4.5.1 Features

- Supports Error Correction for up to 24-bit errors per 551 bytes: User Bootcode 512 bytes + parity 39 bytes and 60-bit errors per 1129 bytes: User Boot code 1024 bytes + parity 105 bytes.
- Supports 512B, 2KB, 4KB, and above as the page size of the NAND flash memory.
- Supports NAND flash memories required RESET command to initialize them.
- Doesn't supports the bad block management.

3.4.5.2 Operation

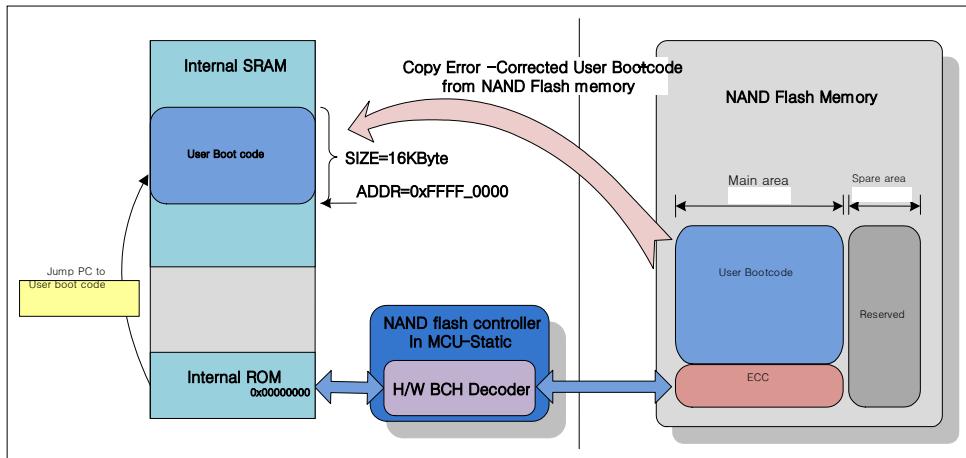


Figure 3-5. NANDBOOTEC Operation

NANDBOOTEC can correct the errors which occur in User bootcode stored in NAND flash memory. Whenever NANDBOOTEC reads User Bootcode from NAND flash memory by 512 bytes or 1024 bytes, it gets to know whether any error of Data exists or not by using Error detection function of H/W BCH decoder included in MCU-S. In case of any error in Data, the maximum 24 or 60 errors can be corrected through H/W Error Correction. If Error Correction fails, Booting with USB Boot mode.

3.4.5.3 How to store User bootcode into the NAND flash memory

Table 3-8 shows the written form of User Bootcode into NAND flash memory. NANDBOOTEC uses the main memory of NAND flash memory and doesn't use the spared area of it.

Sector	Data	Page Size
		512B
0	ECC #0	page #0
1	Bin #0	page #1
2	Bin #1	page #2
3	Bin #2	page #3
4	Bin #3	page #4
5	Bin #4	page #5
6	Bin #5	page #6
7	Bin #6	page #7
8	ECC #1	page #8
9	Bin #7	page #9
10	Bin #8	page #10
11	Bin #9	page #11
12	Bin #10	page #12
13	Bin #11	page #13
14	Bin #12	page #14
15	Bin #13	page #15
16	ECC #1	page #16
17	Bin #14	page #17
18	Bin #15	page #18
19	Bin #16	page #19
20	Bin #17	page #20
21	Bin #18	page #21
22	Bin #19	page #22
23	Bin #20	page #23
24	ECC #2	page #24

ECC #n	64 bytes x 8 = 512 bytes	
	LSB (312 bit)	MSB (200 bit)
LSB 39 bytes	Reserved	
	Parity for sector #(n*8+ 1)	Reserved
	Parity for sector #(n*8+ 2)	Reserved
	Parity for sector #(n*8+ 3)	Reserved
	Parity for sector #(n*8+ 4)	Reserved
	Parity for sector #(n*8+ 5)	Reserved
	Parity for sector #(n*8+ 6)	Reserved
MSB 39 bytes	Parity for sector #(n*8+ 7)	Reserved

25	Bin #21	page #25
26	Bin #22	page #26
27	Bin #23	page #27
28	Bin #24	page #28
29	Bin #25	page #29
30	Bin #26	page #30
31	Bin #27	page #31
32	ECC #3	page #32
33	Bin #28	page #33
34	Bin #29	Page #34
34	Bin #30	page #35
35	Bin #31	page #36

Sector	Data	Page Size			ECC #n	128 bytes x 8 = 1024 bytes		
		2KB	4KB	8K		LSB (840 bit)	MSB (184 bit)	
0	ECC #0	page #0			LSB 105 bytes	Reserved		
1	Bin #0	page #1	page #0			Parity for sector #(n*8+ 1)	Reserved	
2	Bin #1	page #2		page #1		Parity for sector #(n*8+ 2)	Reserved	
3	Bin #2	page #3				Parity for sector #(n*8+ 3)	Reserved	
4	Bin #3	page #4		page #2		Parity for sector #(n*8+ 4)	Reserved	
5	Bin #4	page #5				Parity for sector #(n*8+ 5)	Reserved	
6	Bin #5	page #6		page #3		Parity for sector #(n*8+ 6)	Reserved	
7	Bin #6	page #7				Parity for sector #(n*8+ 7)	Reserved	
8	ECC #1	page #8		page #4				
9	Bin #7	page #9						
10	Bin #8	page #10		page #5				
11	Bin #9	page #11						
12	Bin #10	page #12		page #6				
13	Bin #11	page #13						
14	Bin #12	page #14		page #7				
15	Bin #13	page #15						
16	ECC #1	page #16		page #8				
17	Bin #14	page #17						
18	Bin #15	page #18		page #9				

Table 3-7. NAND Flash memory format for NANDBOOTEC

3.4.6 Additional Information

3.4.6.1 ALIVE Power Control

iROMBOOT changes VDDPWRON and VDDPWRON_DDR pins to High state after Reset in order to supports the fast response to nVDDPWRTOGGLE button. Table 3-9 shows ALIVE module states after iROMBOOT Execution.

Function	State	Description
VDDPWRON	High	Enable Core Power
VDDPWRON_DDR	High	Enable DDR Memory Power
nPADHOLD[2:0]	High	Disable PAD Retention
nPADHOLDEn[2:0]	Low	

Table 3-8. ALIVE Power Control

3.4.6.2 Exception Vector Redirection

Exception Handler of ARM CPU should exist from 0th address by 4 byte one after the other. User generally places the routine jumping to User's Exception Handler to the Exception Handler existing from 0th address. However, in case of iROMBOOT, User's Exception Handler is impossible to set at 0th address because ROM exists at 0th address. CPU Exception can be processed by mapping the arbitrary memory to 0th address when MMU is being used. However

iROMBOOT provides the function redirecting Exception Handler for the System not using MMU.

iROMBOOT uses 32 bytes from the lowest address of internal SRAM as User Exception Vector Table. When Exception occurs, ROM Exception Handler in iROMBOOT lets PC jump to the address of User Exception Handler taken from User Exception Vector Table. Therefore Exception can be processed even at Physical address system by User's setting the address of User Exception Handler to User Exception Vector Table present in internal SRAM and that is equal to High Vector Address.

```
#define BASEADDR_SRAM 0xFFFF0000

//=====
//: Vectors
//=====
.global Vectors
Vectors:
    LDR      pc, ResetV           //: 00 - Reset
    LDR      pc, UndefV          //: 04 - Undefined instructions
    LDR      pc, SWIV             //: 08 - SWI instructions
    LDR      pc, PAabortV         //: 0C - Instruction fetch aborts
    LDR      pc, DAabortV         //: 10 - Data access aborts
    LDR      pc, UnusedV          //: 14 - Reserved (was address
exception)
    LDR      pc, IRQV             //: 18 - IRQ interrupts
    LDR      pc, FIQV             //: 1C - FIQ interrupts

ResetV:
    .word    Reset_Handler

.UndefV:
    .word    (BASEADDR_SRAM + 0x04) //: 04 - undef
.SWIV:
    .word    (BASEADDR_SRAM + 0x08) //: 08 - software interrupt
.PAbortV:
    .word    (BASEADDR_SRAM + 0x0C) //: 0C - prefetch abort
.DAabortV:
    .word    (BASEADDR_SRAM + 0x10) //: 10 - data abort
.UnusedV:
    .word    0
 IRQV:
    .word    (BASEADDR_SRAM + 0x18) //: 18 - IRQ
.FIQV:
    .word    (BASEADDR_SRAM + 0x1C) //: 1C - FIQ

//=====
//: Imports
//=====

.global iROMBOOT
//=====
//: Reset Handler - Generic initialization, run by all CPUs
//=====

Reset_Handler:
```

Figure 3-6. iROMBOOT Exception Handlers

3.4.6.3 Parity generation for Error Correction

NANDBOOTEC can correct the maximum 24 errors in User Bootcode 512 bytes and Parity 39 bytes or maximum 60 errors in User Bootcode 1024 bytes and Parity 105 bytes .Therefore, by generating Parity 39 bytes at every 512 bytes of User Bootcode, User should write the parity information to ECC Sector. For NANDBOOTEC, the number of Parity information varies according to the size of User Bootcode. It is recommended to refer to the description of NANDBOOTEC about the site in which Parity information should be located.

3.4.6.4 CRC32 Error Check

UART and SPI Boot check 16368 (16384 - 16) bytes with CRC32 and if adding CRC32 fcs data to last of transfer of payload then CRC32 fcs will checking will succuss.

CRC32 fcs generator function

```
#define POLY 0x04C11DB7L

unsigned int get_fcs(unsigned int fcs, unsigned char data)
{
    register int i;
    fcs ^= (unsigned int) data;
    for(i=0; i<8; i++)
    {
        if(fcs & 0x01)
            fcs ^= POLY;
        fcs >>= 1;
    }
    return fcs;
}
```

3.4.6.5 Data Decrytion with AES128 ECB mode to use hidden key

All boot mode data will decrypted with AES128 ECB mode to use hidden key by option.

Boot mode sd[4], sdex[4] can select weather data will decrypt or not decrypt.

*Section 4. **System Control***

4.1 Clock Manager

4.1.1 Clock Manager Overview

The clock of the NXP4330D/Q is roughly divided into FCLK, HCLK, MCLK, BCLK and PCLK are used for the ARM CPU core, AXI bus peripherals and APB bus peripherals, respectively. In addition, BCLK is the clock for the NXP4330D/Q system bus. MCLK is the clock for SDRAM memory. The 2-PLL of the NXP4330D/Q is called PLL0 and PLL1, respectively. The 2-PLL and EXTCLK are used to generate the above clocks (i.e. FCLK, HCLK, PCLK, BCLK, MCLK). All PLLs are designed to operate with an X-TAL input of 24MHz.

4.1.2 Clock Manager Features

- Embedded 4-PLL operating independently
- Output Frequency Range
 - PLL0:40M ~ 2.5GHz (non-dithered PLL)
 - PLL1:40M ~ 2.5GHz (non-dithered PLL)
 - PLL2:35M ~ 2.2GHz(dithered PLL)
 - PLL3:35M ~ 2.2GHz(dithered PLL)
- Frequency is changed by Programmable Divider(PDIV, MDIV, SDIV)
- Clock generation for all blocks in the chip
- The PLLs can be switched into Power Down mode by using the program.
- 32.768 KHz supported for Power Management
- Various Power Down Modes
- IDLE mode and STOP mode
- Various Wake Up sources

4.1.2.1 Clock Manager Block Diagram

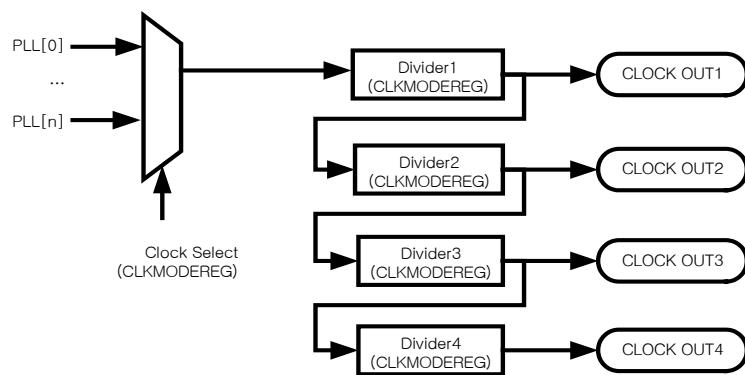


Figure 4-1. Block Diagram of CLOCK MANAGER

The above figure shows a diagram for the clock manager in the NXP4330D/Q. As shown in the above figure, the NXP4330D/Q has four PLLs. The NXP4330D/Q receives the output of PLLs and generates all system clocks, the memory clock and the CPU clock with the output frequency selected among many PLLs.

4.1.3 Clock Manager Functional Description

4.1.3.1 PLL (Phase Locked Loop)

PMS Value

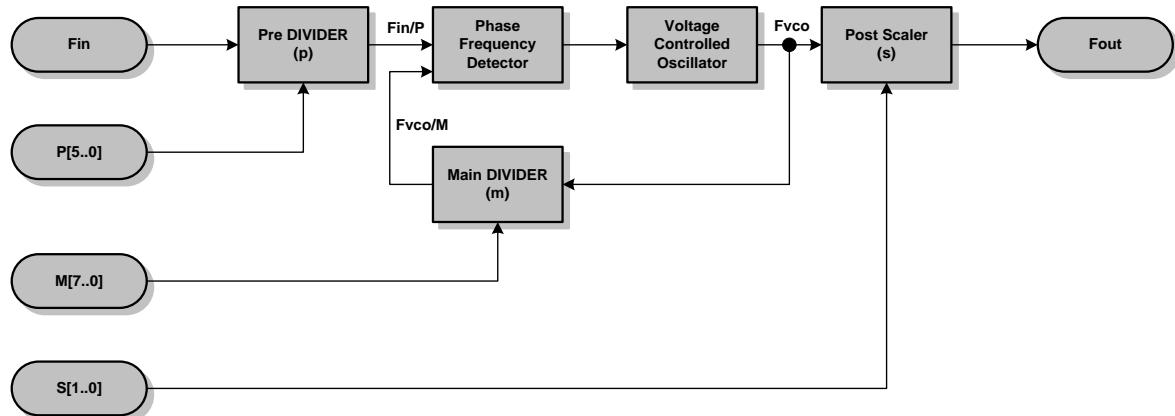


Figure 4-2. Block Diagram of PLL

For the aspect of PLL structure, Figure 4-2 shows the block diagram for one PLL. Fin and Fout indicate input frequency and output frequency respectively. The NXP4330D/Q has PLLs and can generate various programmable clocks by using each PLL.

If the Pre Divider receives a Fin input of 24MHz, it divides the Fin with 'P'. After that, Phase Frequency Detector (PFD) compares the difference between Fin/P (Reference Clock) and Fvco/M (Feedback Clock). The amplitude of the voltage varies depending on the difference of the values compared between Fin/P and Fvco/M. If the reference clock is faster than the feedback block, the Voltage Controlled Oscillator (VCO) increases in proportion to the difference. If the reference clock is delayed than the feed clock, VCO is decreased, and it generates an Fvco clock. That is because VCO is a voltage value and plays the role of controlling the clock speed to be faster or slower. At this point, the voltage value is determined by the difference of the values compared between the reference clock and the feedback clock. If the Fvco is not a desired clock, the feedback is recreated through the Main Divider and compared in PFD. These steps are repeated until the reference clock and the feedback clock become equal. If proper FVCO is out, the final Fout clock is created as the divide value(s) of a Post Scaler. Finally, the desired clock frequency is determined by the p, m and s values.

As described above, Fout can be variously set by Fin and p/m/s values and the equation to specify p/m/s values is as follows: (Note that all PLL0/1/2/3 indicate Fout. Equation may vary for each case.)

$$\text{PLL } x = (m * \text{Fin}) / (p * 2^s)$$

(x = 0,1,2,3, m = MDIV, p = PDIV, s = SDIV = 0, 1, 2, 3)

The range of the MDIV and PDIV values for PLL x are as follows:

Range of MDIV Value : $64 \leq \text{MDIV} \leq 1023$

Range of PDIV Value : $1 \leq \text{PDIV} \leq 63$

The PDIV and the MDIV values should be selected by considering the VCO value and NXP4330D/Q's stable operation.

The NXP4330D/Q has PLLs and each PLL has different default values and operation ranges. The basic frequencies for the NXP4330D/Q are listed in the table below:

PLL	INITIAL FREQUENCY	RECOMMENDED FREQUENCY (Fvco)	RECOMMENDED FREQUENCY (fout)	INITIAL PDIV/ MDIV/ SDIV VALUE		
				PDIV	MDIV	SDIV
PLL0	550.000000Mhz	1250 ~ 2500Mhz	40 ~ 2500Mhz	6	550	2
PLL1	147.456 MHz	1250 ~ 2500Mhz	40 ~ 2500Mhz	6	590	4
PLL2	96Mhz	40~2200Mhz	1100~2200Mhz	3	192	4
PLL3	125Mhz	40~2200Mhz	1100~2200Mhz	3	250	4

Table 4-1. Initial PDIV/ MDIV/ SDIV Value

For all blocks except for CPU, the operation status (Run/Stop) of the memory controller should be checked before changing the PLL output frequency. In addition, the PLL change bit (**PWRMODE.CHGPLL**) should be set as '1' after PLL change (**PLLSETREG0, PLLSETREG1**).

Setting guide of PMSK

- p, m, s and k are decimal values of P[5:0], M[8:0], S[2:0] and K[15:0], respectively.
 - $p = P[5:0]$, $m = M[8:0]$, $s = S[2:0]$, $k = K[15:0]$
 - F_{FVCO} and F_{FOUT} are calculated by the following equation.
 - $F_{FVCO} = ((m+k/65536) \square F_{FIN}) / p$
 - $F_{FOUT} = ((m+k/65536) \square F_{FIN}) / (p \times 2^s)$
- While range of registers P[5:0], M[8:0] and S[2:0] are unsigned integers, K[15:0] is a two's complement integer.
 - $6'b00\ 0001 \leq P[5:0] \leq 6'b11\ 1111$ and $2MHz \leq F_{FREF}(F_{FIN} / p) \leq 30MHz$
 - $9'b0\ 0100\ 0000 \leq M[8:0] \leq 9'b1\ 1111\ 1111$
 - $3'b000 \leq S[2:0] \leq 3'b101$
 - $16'b1000\ 0000\ 0000\ 0000 \leq K[15:0] \leq 16'b0111\ 1111\ 1111\ 1111$
- Setting P[5:0] or M[8:0] to all zeros is strictly prohibited while RESETB is logic high. (6'b00 0000 / 9'b0 0000 0000)
- The division ratio of scaler is controlled by S[2:0] as summarized in Table 4-2.
- Setting S[2:0] to the values in the gray rows in Table 4-2 is strictly prohibited.

S[2:0]	Division Ratio
000	$2^0=1$
001	$2^1=2$
010	$2^2=4$
011	$2^3=8$
100	$2^4=16$
101	$2^5=32$
110	Prohibited
111	Prohibited

Table 4-2. Division ratio of scaler

Setting guide of SSCG_EN, SEL_PF, MFR and MRR

- When SSCG_EN is set to logic high, the spread spectrum mode is enabled.
- sel_pf, mfr and mrr are decimal values of SEL_PF[1:0], MFR[7:0] and MRR[5:0], respectively.
 - $sel_pf = SEL_PF[1:0]$, $mfr = MFR[7:0]$, $mrr = MRR[5:0]$
- Modulation frequency, MF, is determined by the following equation.
 - $MF = F_{FIN} / p / mfr / 2^5 [Hz]$
- Modulation rate(pk-pk), MR, is determined by the following equation.
 - $MR = mfr \times mrr / m / 2^6 \times 100 [\%]$
- Modulation mode is determined by sel_pf.
 - 00: down spread, 01: up spread, 1x: center spread
- Range of registers.
 - $8'b0000\ 0000 \leq MFR[7:0] \leq 8'b1111\ 1111$
 - $6'b00\ 0001 \leq MRR[5:0] \leq 6'b11\ 1111$
 - $0 \leq mrr \times mfr \leq 512$
 - $2'b00 \leq SEL_PF[1:0] \leq 2'b10$

PDIV/ MDIV/ SDIV Values for PLL0, PLL1

Input Frequency	Output Frequency (MHz)	PDIV/ MDIV/ SDIV Value			Remark
		PDIV	MDIV	SDIV	
24MHz	2000.000000	6	500	0	Maximum Available Frequency
24MHz	1900.000000	6	475	0	
24MHz	1800.000000	4	300	0	
24MHz	1700.000000	6	425	0	
24MHz	1600.000000	6	400	0	
24MHz	1500.000000	4	250	0	
24MHz	1400.000000	6	350	0	
24MHz	1300.000000	6	325	0	
24MHz	1200.000000	4	400	1	
24MHz	1100.000000	6	550	1	
24MHz	1000.000000	6	500	1	
24MHz	900.000000	4	300	1	
24MHz	800.000000	6	400	1	Maximum Available Frequency
24MHz	780.000000	4	260	1	
24MHz	760.000000	6	380	1	
24MHz	740.000000	6	370	1	
24MHz	720.000000	4	240	1	
24MHz	562.000000	6	562	2	
24MHz	533.000000	6	533	2	

24MHz	490.000000	6	490	2	
24MHz	470.000000	6	470	2	
24MHz	460.000000	6	460	2	
24MHz	450.000000	4	300	2	
24MHz	440.000000	6	440	2	
24MHz	430.000000	6	430	2	
24MHz	420.000000	4	280	2	
24MHz	410.000000	6	410	2	
24MHz	400.000000	6	400	2	
24MHz	399.000000	4	266	2	
24MHz	390.000000	4	260	2	
24MHz	384.000000	4	256	2	
24MHz	350.000000	6	350	2	
24MHz	330.000000	4	220	2	
24MHz	300.000000	4	400	3	
24MHz	266.000000	6	532	3	
24MHz	250.000000	6	500	3	
24MHz	220.000000	6	440	3	
24MHz	200.000000	6	400	3	
24MHz	166.000000	6	332	3	
24MHz	147.45600	6	590	4	147.5MHz (0.03% error)
24MHz	133.000000	6	532	4	
24MHz	125.000000	6	500	4	
24MHz	100.000000	6	400	4	
24MHz	96.000000	4	256	4	
24MHz	48.000000	3	96	4	

Table 4-3. PDIV/ MDIV/ SDIV Value for PLL0

PDIV/ MDIV/ SDIV Values for PLL2, PLL3

Input Frequency	Output Frequency (MHz)	PDIV/ MDIV/ SDIV Value			Remark
		PDIV	MDIV	SDIV	
24MHz	2000.000000	3	250	0	Maximum Available Frequency
24MHz	1900.000000	3	238	0	
24MHz	1800.000000	3	225	0	
24MHz	1700.000000	3	213	0	
24MHz	1600.000000	3	200	0	
24MHz	1500.000000	4	250	0	
24MHz	1400.000000	3	175	0	
24MHz	1300.000000	3	163	0	

24MHz	1200.000000	3	150	0	
24MHz	1100.000000	3	275	1	
24MHz	1000.000000	3	250	1	
24MHz	900.000000	3	225	1	
24MHz	800.000000	3	200	1	Maximum Available Frequency
24MHz	780.000000	3	195	1	
24MHz	760.000000	3	190	1	
24MHz	740.000000	3	185	1	
24MHz	720.000000	3	180	1	
24MHz	562.000000	3	141	1	
24MHz	533.000000	3	267	2	
24MHz	490.000000	3	245	2	
24MHz	470.000000	3	235	2	
24MHz	460.000000	3	230	2	
24MHz	450.000000	3	225	2	
24MHz	440.000000	3	220	2	
24MHz	430.000000	3	215	2	
24MHz	420.000000	3	210	2	
24MHz	410.000000	3	205	2	
24MHz	400.000000	3	200	2	
24MHz	399.000000	4	266	2	
24MHz	390.000000	3	195	2	
24MHz	384.000000	3	192	2	
24MHz	350.000000	3	175	2	
24MHz	330.000000	3	165	2	
24MHz	300.000000	3	150	2	
24MHz	266.000000	3	266	3	
24MHz	250.000000	3	250	3	
24MHz	220.000000	3	220	3	
24MHz	200.000000	3	200	3	
24MHz	166.000000	3	166	3	
24MHz	147.45600	3	147	3	
24MHz	133.000000	3	266	4	
24MHz	125.000000	3	250	4	
24MHz	100.000000	3	200	4	
24MHz	96.000000	3	192	4	
24MHz	48.000000	3	96	4	

Table 4-4. PDIV/ MDIV/ SDIV Value for PLL1

PLL Power Down

The NXP4330D/Q supports PLL Power Down mode to minimize power consumption. For example, if all system clocks are generated with PLL0 and PLL1 does not need to be used. Therefore, power does not need to be supplied to the PLL1. In such a case, the NXP4330D/Q switches PLL1 into power down mode to reduce the power consumption. However, PLL0 cannot enter to the power down mode. PLL0 power down can be achieved by writing '1' to the **CLKMODEREG0.PLLPWDN1**.

4.1.3.2 Change PLL Value

When CPU want to change the PLL divider value. The PLL Change Bit (**PWRMODE.CHGPLL** bit) must be set to 1 after setting the PLL Setting Reset (**PLLSETREG0**, **PLLSETREG1**) to appropriate value.

Power management and Clock Controller blocks up the clock supplied to internal controllers because PLLs are unstable when PLL divider value is changed. After locking time, these blocks supply clock. CPU must check whether the blocks run or stop such as STOP mode.

4.1.3.3 Clock Generator

Clocks Summary

The 5 clocks created in the NXP4330D/Q and the maximum frequencies for each clock are listed in the table below. The minimum frequency is not limited within the clock frequency limit creatable in PLL.

Clock Name	Min Frequency	Max Frequency (MHz)	Description
FCLKCPU0	-	800/1000 *Note1	CPU CLOCK
HCLKCPU0	-	250	CPU BUS CLOCK
MDCLK		800	Memory DLL clock.
MCLK	Note	800	Memory clock. Note*. Minimum frequency of MCLK is determined by SDRAM specification.
MBCLK		400	Memory BUS CLOCK(MCU CLOCK)
MPCLK		200	Memory Peripheral clock.
BCLK	-	333MHz	SYSTEM BUS CLOCK(CORE CLOCK) CORE blocks operates on the basis of BCLK. (MPEG, DMA, etc...)
PCLK	-	166MHz	PERIPHERAL BUS CLOCK CPU accesses a block register via I/O with PCLK.
GR3DBCLK		333	GPU clock
GR3DPCLK			Not used.
MPEGBCLK		300	MFC clock. (BUS and CODEC)
MPEGPLK		150	MFC clock (Peripheral clock)

Table 4-5. NXP4330D/QClock Summary

In Table 4-5, note that the size of PCLK should be the half size of the BCLK when the maximum/minimum frequency values are specified.

* Note1 : CPU frequency is 800Mhz at 1.0V and 1000Mhz at 1.1V

CPU0 Clock

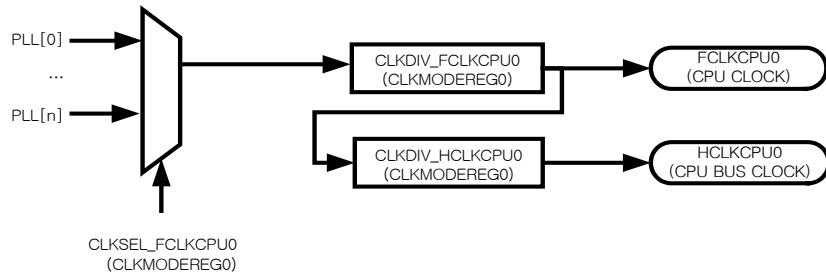


Figure 4-3. CPU Clock

Figure 4-3 shows a block diagram that creates the clock supplied to FCLKCPU0, which is the main CPU of the NXP4330D/Q. **CLKMODEREG0** selects a desired PLL output from among PLLs. With the clock created from the selected PLL, the **CLKDIV_FCLKCPU0** register and **CLKDIV_HCLKCPU0** generates **FCLKCPU0** to be supplied to the core block of CPU and **HCLKCPU** to be supplied to the AXI bus clock. Be careful not to set HCLKCPU over maximum speed. The frequency of FCLKCPU and HCLKCPU cannot be the same.

Any PLL can be used to generate the CPU clock, but it is recommended to use the PLL0

Recommended clock frequency as follows:

CPU Operation Voltage.	FCLKCPU0 (Mhz)	HCLKCPU0 (Mhz)
1.0V	800	200
1.1V	1000	250

System Bus Clock (Core clock)

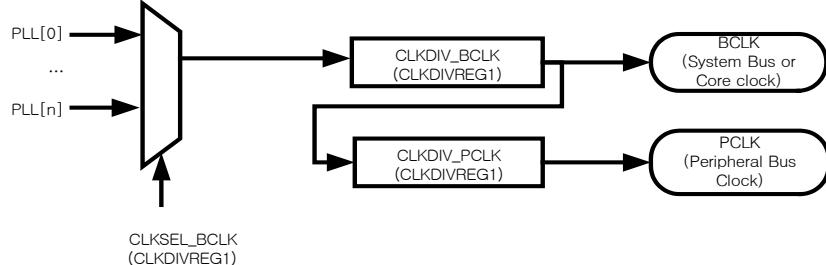


Figure 4-4. System BUS Clock

System bus clock (BCLK) is used as Core clock. The system bus clock is called [BCLK] and the half clock of BCLK is called [PCLK]. BCLK is the clock for all SOC Core operations. PCLK is used when the CPU accesses each block register via I/O. Therefore, PCLK should not be applied to the blocks not being used. Every block has PCLK enable/disable Register. The blocks that PCLK is applied to have (refer to each Section). These registers decide if PCLK is applied to a block only when the CPU accesses the corresponding block register or when it is always applied.

clock frequency ratio should be as follows.

BCLK : PCLK = 2 : 1

Recommended clock frequency as follows:

mode	BCLK (Mhz)	PCLK (Mhz)
max operation.	333	166

Memory Bus Clock (MCU Clock)

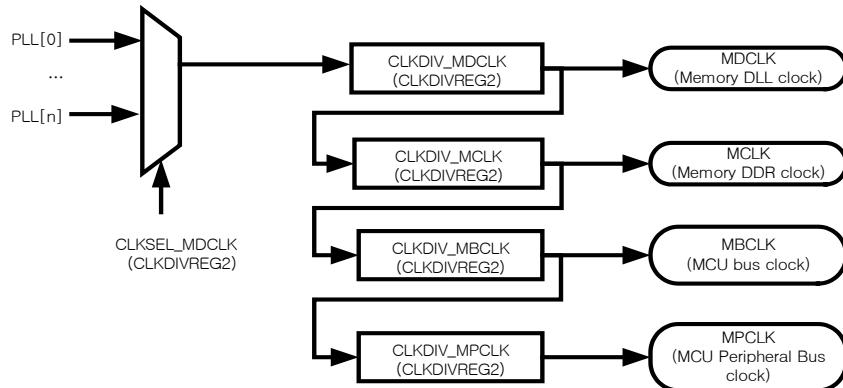


Figure 4-5. System BUS Clock

Memory bus clock (MCLK) is used as SDRAM, MCU Core clock. MDCLK is the clock for DLL of Memory Control Unit(MCU). MCLK is the DDR interface clock. MBCLK is bus clock of MCU. MPCLK is peripheral bus clock of MCU. The MCLK frequency should be the doublethat of the MBCLK frequency. (BCLK to MCLK is a 1:2 ratio)

Recommended clock frequency ratio is as follows.

MDCLK : MCLK : MBCLK : MPCLK = 4 : 4 : 2 : 1

Recommended clock frequency as follows:

mode	MDCLK (Mhz)	MCLK (Mhz)	MBCLK (Mhz)	MPCLK (Mhz)
Fast	800 (Note*)	800	400	200
Slow	800	200	100	50

*Note: MDCLK should be divided by 2, i.e. CLKDIV_MDCLK should be 1.

GPU (Graphic Processing Unit) clock

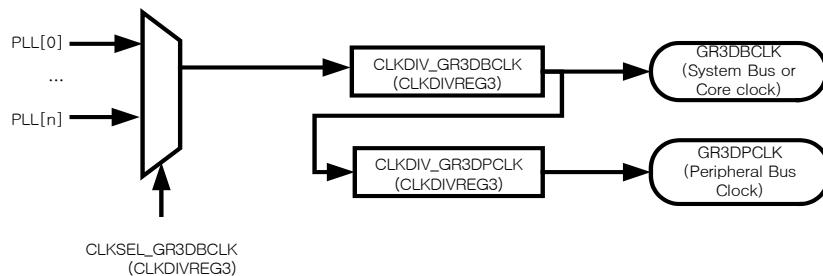


Figure 4-6. System BUS Clock

GPU clock (GR3DBCLK) is used as GPU core clock. GR3DPCLK is not used (reserved).

Recommended clock frequency as follows:

mode	GR3DBCLK (Mhz)	GR3DPCLK (Mhz)
Max operation.	333	166 (not used)

MFC(Multi Function Codec) clock

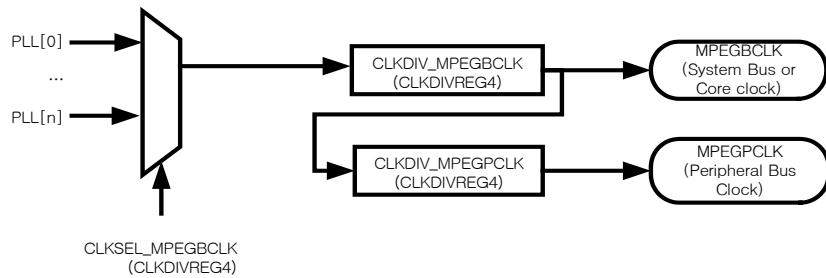


Figure 4-7. System BUS Clock

MFC clock (MPEGBCLK) is used as Multi function codec clock. MPEGPCLK is used peripheral bus clock for MFC unit.

Recommended clock frequency as follows:

mode	MPEGBCLK (Mhz)	MPEGPCLK (Mhz)
Max operation.	300	150

4.2 Power Manager

4.2.1 Power Manager Overview

The power manager of the NXP4330D/Q provides the following functions to operate the system stably and reduce the power consumption.

- Power Up Sequence
- Reset Generation
- Power Management
- Change PLL Value

The key functions of the power manager are to control the Power up Sequence to make the NXP4330D/Q stable after the power is supplied to the system and to manage the power effectively. Apart from this, it controls the reset configuration in initial operation.

In addition, this block generates various reset signals, such as External Reset Output (nRSTOUT), AliveGPIO Reset and Soft Reset.

- The NXP4330D/Q provides various Power Down modes to reduce the system power consumption. The three Power modes provided by the NXP4330D/Q are as follows: Normal Mode
 - IDLE Mode
 - STOP Mode
 - SLEEP Mode (See the 'ALIVE' Section for SLEEP Mode)

4.2.2 Power Down Mode Operation

Figure 4-8 shows the state diagram for the Power Management Block. The figure indicates the entry conditions for each Power Down mode and all Wake Up conditions.

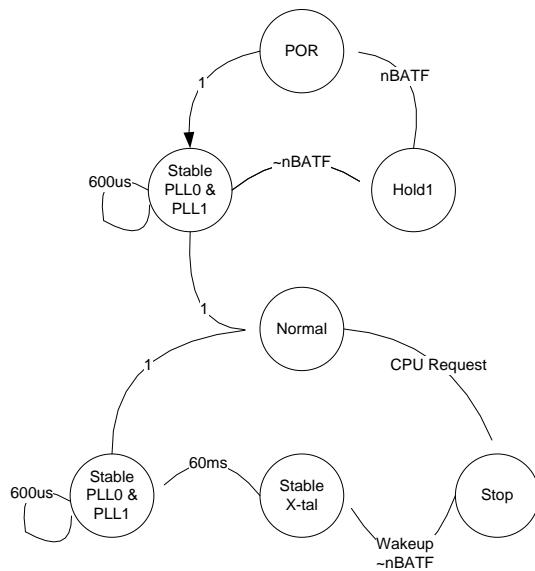


Figure 4-8. Power Management Sequence

<NXP4330D/Q State>

- POR : Power On Reset State
- StablePLL : Wait for PLL locking time
- NORMAL : Normal Operation State
- STOP : Stop Operation Mode
- StableX-tal : Wait for Crystal's stable oscillation
- Hold : Wait for nBATF= High.

<Wake Up Source>

- SWRST : Software Reset
- SWRSTENB : Software Reset Enable
- ALIVEGPIOEvent : ALIVE GPIO Wake Up Event
- CUIIRQ : Interrupt from CPU(IDLE Mode)
- RTCIRQ : Interrupt from RTC
- BAFT : Battery Fault
- VDDPWRTOGGLE : VDDPWRTOGGLE Switch Push Button
- WRST : Watchdog Reset

4.2.2.1 IDLE Mode

In the IDLE mode, since the power and clocks are supplied to all blocks except for the CPU clock, power consumption can be reduced a bit. To enter to IDLE mode, the **PWRMODE.IDLE** Register should be set as “1”. In IDLE Mode, the CPU clock is not supplied, but the power is normally supplied and PLLs operate normally.

Wake-Up Source can use all the NXP4330D/Q interrupts that can be generated by the Interrupt controller: GPIO Interrupt, Alive GPIO Interrupt, External Interrupt and RTC Interrupt. The interrupt for the Wake-Up Source should be

enabled before entering to the IDLE Mode. The CPU returns to the previous status immediately after it is woken up in IDLE Mode.

4.2.2.2 STOP Mode

In STOP mode, the clock is not supplied to all blocks including the ARM Core, because the PLL also does not operate in the clock controller if the clock is not supplied to all blocks. However, the NXP4330D/Q converts DRAM into Self Refresh mode to protect memory data before entering to STOP mode. Like IDLE mode, the **PWRMODE .STOP** should be set as '1' to enter to STOP mode.

The Wake Up source is slightly limited in STOP mode. The available Wake Up sources are RTC Interrupt, Alive GPIO Interrupt, etc. The Wake Up source is limited because the clock is not supplied to all the other blocks except for the power manager and RTC block. Since the RTC block uses a separate power and clock, only interrupts by the RTC clock can be used as a Wake Up source.

Unlike with IDLE mode, all PLLs stop when the system is woken up in STOP mode so that the system cannot return to the previous status, immediately. Therefore, the Wake Up time is about 70 ms longer than that of IDLE mode to stabilize the internal PLLs.

Power Down Mode	Power Supply	CPU Clock Supply	Other Clock Supply	SDRAM Mode	Wake Up Condition
IDLE MODE	ON	OFF	ON	NORMAL	RTC Interrupt, AliveGPIO Interrupt, All Interrupt to Interrupt Controller, External IRQ
STOP MODE	ON	OFF	OFF	Self Refresh	RTC Interrupt, AliveGPIO Interrupt

Table 4-6. Wake Up Condition and Power Down Mode Status

4.2.2.3 SLEEP Mode 1, SLEEP Mode 2

See the 'ALIVE' Section for SLEEP Mode1 and SLEEP Mode2.

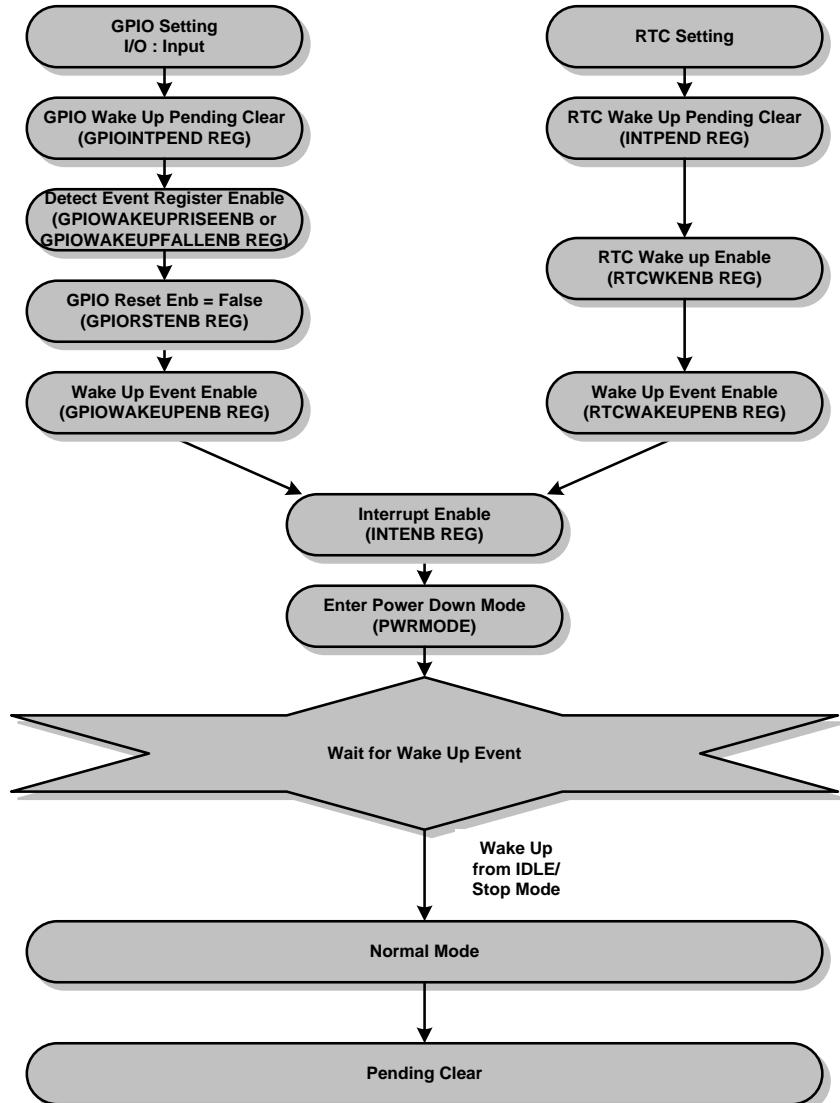


Figure 4-9. Power Down Mode Sequence

Figure 4-9 shows the sequence to enter the Power Down mode and the Wake Up procedure. First, the Wake Up source selects a desired event (interrupt) and specifies the attribute of the event. If the Wake up Source is GPIO, the setting is changed into Input and the Pending Clear is performed. In addition, the status to detect that an event (interrupt) is specified and Software Reset Enb is set as False for the worst case (If the Software Reset Enb switch is not implemented in terms of Hardware, False does not need to be specified). Finally, the system enables the relevant interrupt (if an interrupt is used) and enters a Power Down mode.

In this Power Down mode, the NXP4330D/Q waits a Wake Up event (interrupt). If the Wake Up event (Interrupt) occurs, the NXP4330D/Q returns to normal mode and clears the relevant interrupt pending in terms of Software.

4.2.2.4 GPIO as a Wake up Source

GPIO is available for all Power Down modes. However, since the internal power and the clock status are not equal for each power modes, the operation status is a little different at each power mode.

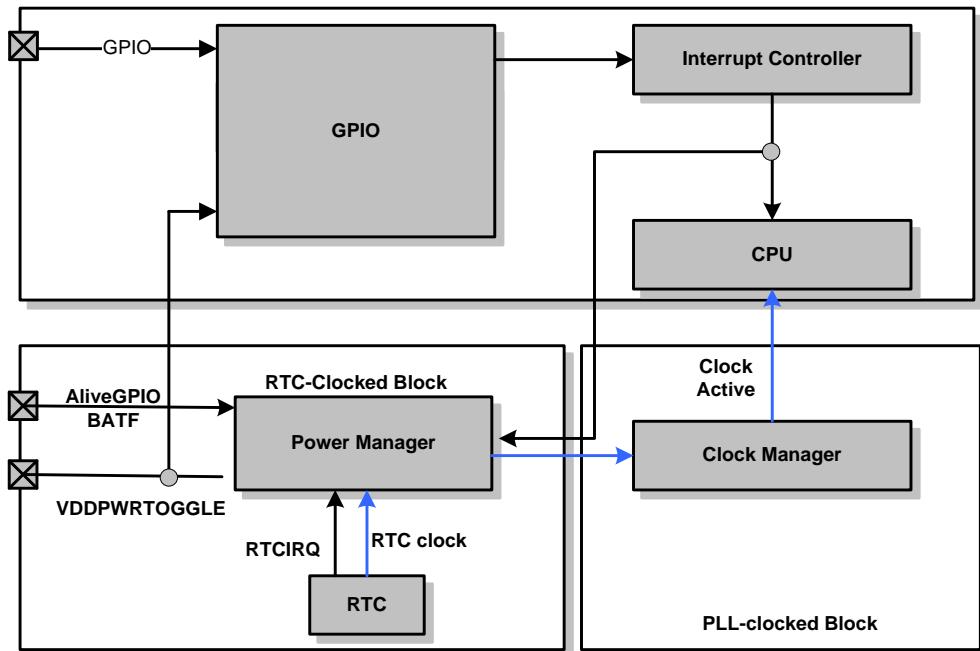


Figure 4-10. Wake Up Block Diagram

As shown in Figure 4-10, Power Manager use different clock. Since RTC-clock is always supplied to Power Manager Block, the PADs can be used as a Wake Up Source.

The description of the Wake Up procedures in Power Down mode is as follows:

- Wake Up in IDLE Mode

During IDLE mode, clock and the power for the other blocks are supplied normally except CPU clock. Therefore, the input received in GPIO is applied to the interrupt controller and wakes up the CPU.

- Wake Up in STOP Mode

In STOP mode, all clocks except for the RTC clock are not supplied (PLL and XTI are included). The interrupt controller does not operate in STOP mode. At this time, if a signal is entered to Power Manager, the power manager wakes up the clock manager, first. As a result of this Wake Up, all clocks, such as the PLL and PCLK, BCLK, MCLK and FCLK, are enabled and supplied to the CPU and the whole system. In other words, the system is woken up. The Wake Up time is about 70 ms longer than that of IDLE mode to stabilize the PLLs.

4.3 Reset Generation

4.3.1 Power On Reset Sequence

Power management block has the reset generation block. The reset generation block uses the nPORST which is sampled at RTC clock (32.768 KHz). And the RTC clock is used as main clock for power management. So Even if the RTC Function is not used, the RTC clock must be supplied.

Figure 4-11 shows the clock and reset behavior during the power-on reset sequence.

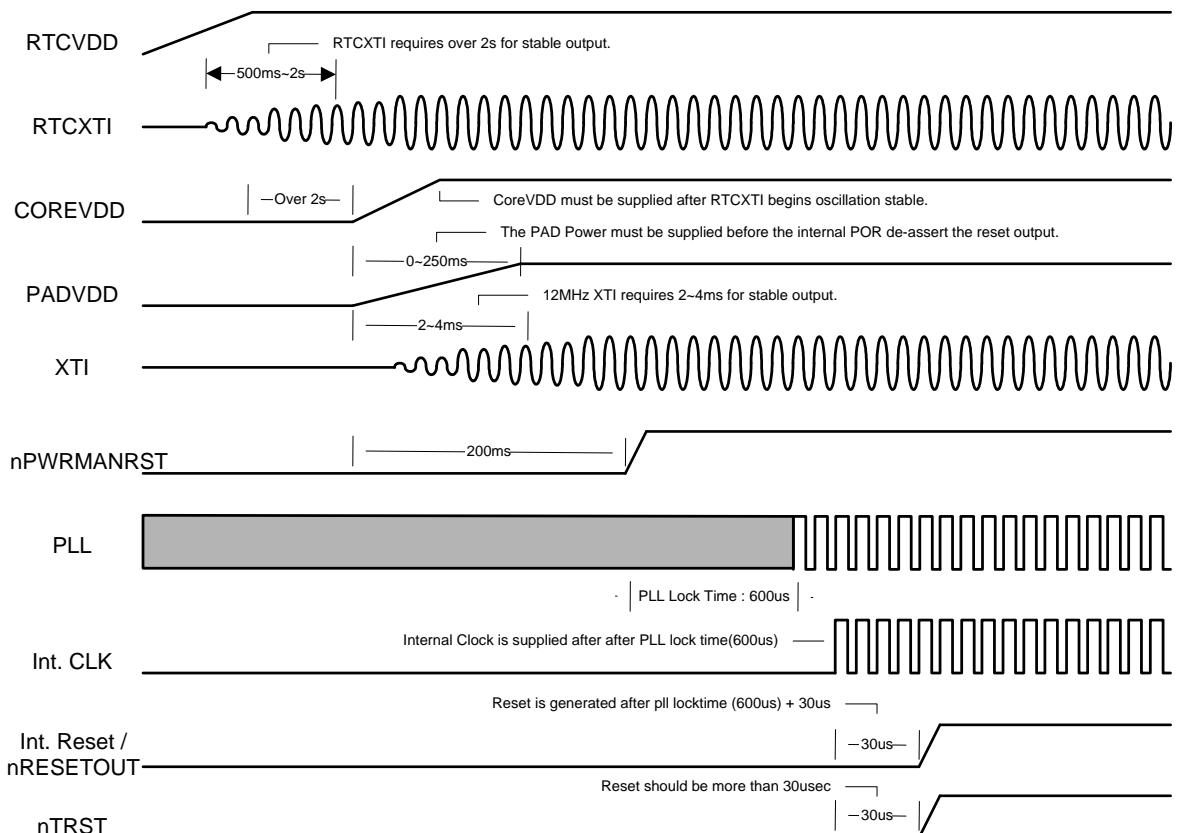


Figure 4-11. Power-On Reset Sequence

nPORST (Power on reset) PAD must be hold to Low (GND) while the crystal oscillator begins stable oscillation after power on. It takes a least 60ms to stabilize oscillations. When nPORST is released after the stabilization of OSC pad (XTI, RTCXTI), the internal PLL start to operate according to the default PLL configuration. But PLL needs the locking time before the stable clock is output. While locking time, power manage doesn't supply the clock for internal logic. After PLL clock is stable, PLL clock output is supplied for each block, then The internal reset and external reset are released after 30us.

4.3.2 Software Reset and GPIO Reset

NXP4330D/Q supports Software Reset that CPU can reset itself with Software reset. To generate Software Reset, **SWRSTENB** bit must be set to 1 before setting **PWRMODE.SWRST** bit. Software Reset mode does not need the time for stabilization clock because the software reset is requested in stable state differently from power on reset.

NXP4330D/Q supports user defined GPIO Reset. The Power management block generates reset when the AliveGPIO pad defined as GPIO Reset source is asserted or de-asserted. The AliveGPIO pad is used as GPIO Reset source are defined at Wakeup Source Register. The GPIORSTENB bit set to 1 enables the AliveGPIO Reset source feature.

4.3.3 Watchdog Reset

The Watchdog timer block is used to resume the controller operation whenever it is disturbed by malfunctions such as system error, etc. When power management block detects the event from watchdog timer, it generates exactly the same reset as power on reset because the watchdog reset event occurs in malfunctions and unknown state.

4.3.3.1 nPORST, Software Reset, Watchdog Reset and GPIO Reset

NXP4330D/Q has four reset states as below.

Blocks	Power On Reset	Watchdog Reset	GPIO Reset (Software Reset)	Wake Up (Idle, Stop)
Clock Manager	Reset	Reset	Reset	X
All Core(CPU and etc...)	Reset	Reset	Reset	X
GPIO	Reset	Reset	Reset	X
Power Manager (Except LASTPWRMODE Register)	Reset	Reset	Reset	X
LASTPWRMODE Register	Reset	X	X	X
RTC Registers (Except RTCCNTREAD Register)	Reset	Reset	Reset	X
RTCCNTREAD Register	X	X	X	X
nGRESETOUT (Output to PAD)	Reset	Reset	Reset	X

Table 4-7. Reset State

4.4 Register Summary

Bit	R/W	Symbol	Description	Reset Value
CLOCK MODE REGISTER0 (CLKMODEREG0)				
<i>Address : C001_0000h</i>				
[31]	R/W	WAIT_UPDATE_PLL	Wait flag updating-PLL for several RTC(32768Hz) clocks 1=in-process , 0=done.	1'b0
[30:4]	-	RESERVED	reserved	1'b0
[3]	R/W	UPDATE_PLL[3]	Update P,M,S values for PLL[3] 1=update, 0=none..	1'b0
[2]	R/W	UPDATE_PLL[2]	Update P,M,S values for PLL[2] 1=update, 0=none..	1'b0
[1]	R/W	UPDATE_PLL[1]	Update P,M,S values for PLL[1] 1=update, 0=none..	1'b0
[0]	R/W	UPDATE_PLL[0]	Update P,M,S values for PLL[0] 1=update, 0=none..	1'b0
<Note>The PMS values of PLL[n] is applied when UPDATE_PLL[n] is set to '1'.				
CLOCK MODE REGISTER1 (CLKMODEREG1)				
<i>Address : C001_0004h</i>				
[31:0]	-	RESERVED	Reserved	1'b0
PLL0 SETTING REGISTER (PLLSETREG0)				
<i>Address : C001_0008h</i>				
[31]	-	RESERVED	Reserved	1'b0
[30]	R/W	SSCG_EN	PLL spread spectrum enable.This register is ignored for non-dithered PLL 1=enable, 0=disable	1'b0
[29]	R/W	PD	PLL Power down 1=power down, 0=power on.	1'b0
[28]	R/W	NPLLBYPASS	This register bypass PLL outputs. 1: normal PLL output 0 :Xrystal clock (PLL input) is selected as PLL output.	1'b1
[27:24]	R/W	PLOUTDIV	PLL output for peripheral is divided by this register. PLL output for system clock is not affected by this register. 0=divide by one, 1=divide by two, N-1=divide by N	4'b0
[23:18]	R/W	PDIV	Pre Divider Value. Divider initially dividing 12 MHz to make PLL0 $1 \leq PDIV \leq 63$	6'd6
[17:8]	R/W	MDIV	Main Divider Value. If the value of Fvco is not a desired clock, the value is divided into MDIV again after feedback. (This loop is continuously circulated until a desired clock frequency is made.) $64 \leq MDIV \leq 1023$	10'd550
[7:0]	R/W	SDIV	Post Scale Divider. Divide the value of Fvco by SDIV to obtain the desired PLL0 value finally. $0 \leq SDIV \leq 5$	8'd2
PLL1 SETTING REGISTER (PLLSETREG1)				
<i>Address : C001_000Ch</i>				
[31]	-	RESERVED	Reserved	1'b0

Bit	R/W	Symbol	Description	Reset Value
[30]	R/W	SSCG_EN	PLL spread spectrum enable. This register is ignored for non-dithered PLL 1=enable, 0=disable	1'b0
[29]	R/W	PD	PLL Power down 1=power down, 0=power on.	1'b0
[28]	R/W	NPLLBYPASS	This register bypass PLL outputs. 1: normal PLL output 0:Xrystal clock (PLL input) is selected as PLL output.	1'b1
[27:24]	R/W	PLLOUTDIV	PLL output for peripheral is divided by this register. PLL output for system clock is not affected by this register. 0=divide by one, 1=divide by two, N-1=divide by N	4'b0
[23:18]	R/W	PDIV	Pre Divider Value. Divider initially dividing 12 MHz to make PLL0 1 ≤ PDIV ≤ 63	6'd6
[17:8]	R/W	MDIV	Main Divider Value. If the value of Fvco is not a desired clock, the value is divided into MDIV again after feedback. (This loop is continuously circulated until a desired clock frequency is made.) 64≤ MDIV ≤ 1023	10'd590
[7:0]	R/W	SDIV	Post Scale Divider. Divide the value of Fvco by SDIV to obtain the desired PLL0 value finally. 0 ≤ SDIV ≤ 5	8'd4

PLL2 SETTING REGISTER (PLLSETREG2)

Address : C001_0010h

[31]	-	RESERVED	Reserved	1'b0
[30]	R/W	SSCG_EN	PLL spread spectrum enable. This register is ignored for non-dithered PLL 1=enable, 0=disable	1'b0
[29]	R/W	PD	PLL Power down 1=power down, 0=power on.	1'b0
[28]	R/W	NPLLBYPASS	This register bypass PLL outputs. 1: normal PLL output 0:Xrystal clock (PLL input) is selected as PLL output.	1'b1
[27:24]	R/W	PLLOUTDIV	PLL output for peripheral is divided by this register. PLL output for system clock is not affected by this register. 0=divide by one, 1=divide by two, N-1=divide by N	4'b0
[23:18]	R/W	PDIV	Pre Divider Value. Divider initially dividing 12 MHz to make PLL0 1 ≤ PDIV ≤ 63	6'd3
[17:8]	R/W	MDIV	Main Divider Value. If the value of Fvco is not a desired clock, the value is divided into MDIV again after feedback. (This loop is continuously circulated until a desired clock frequency is made.) 64≤ MDIV ≤ 1023	10'd192
[7:0]	R/W	SDIV	Post Scale Divider. Divide the value of Fvco by SDIV to obtain the desired PLL0 value finally. 0 ≤ SDIV ≤ 5	8'd4

PLL3 SETTING REGISTER (PLLSETREG3)

Address : C001_0014h

[31]	-	RESERVED	Reserved	1'b0
[30]	R/W	SSCG_EN	PLL spread spectrum enable. This register is ignored for non-dithered PLL 1=enable, 0=disable	1'b0

Bit	R/W	Symbol	Description	Reset Value
[29]	R/W	PD	PLL Power down 1=power down, 0=power on.	1'b0
[28]	R/W	NPLLBYPASS	This register bypass PLL outputs. 1: normal PLL output 0 :Xystal clock (PLL input) is selected as PLL output.	1'b1
[27:24]	R/W	PLLOUTDIV	PLL output for peripheral is divided by this register. PLL output for system clock is not affected by this register. 0=divide by one, 1=divide by two, N-1=divide by N	4'b0
[23:18]	R/W	PDIV	Pre Divider Value. Divider initially dividing 12 MHz to make PLL0 $1 \leq PDIV \leq 63$	6'd3
[17:8]	R/W	MDIV	Main Divider Value. If the value of Fvco is not a desired clock, the value is divided into MDIV again after feedback. (This loop is continuously circulated until a desired clock frequency is made.) $64 \leq MDIV \leq 1023$	10'd250
[7:0]	R/W	SDIV	Post Scale Divider. Divide the value of Fvco by SDIV to obtain the desired PLL0 value finally. $0 \leq SDIV \leq 5$	8'd4
RESERVED				
<i>Address : C001_0018h ~ C001_001Fh</i>				
FCLKCPU0 SETTING REGISTER (CLKDIVREG0)				
<i>Address : C001_0020h</i>				
[31:27]	R/W	RESERVED	Reserved	4'b0
[26:21]	R/W	RESERVED	Reserved	6'b0
[20:15]	R/W	RESERVED	Reserved	6'b1
[14:9]	R/W	CLKDIV_HCLKCPU0	Divide value to create the clock of HCLKCPU0. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[8:3]	R/W	CLKDIV_FCLKCPU0	Divide value to create the clock of FCLKCPU0. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[2:0]	R/W	CLKSEL_FCLKCPU0	Select a clock source. If this bits is N, then PLL[N] is selected as clock source.	3'b0
BCLK SETTING REGISTER (CLKDIVREG1)				
<i>Address : C001_0024h</i>				
[31:27]	R/W	RESERVED	Reserved	4'b0
[26:21]	R/W	RESERVED	Reserved	6'b0
[20:15]	R/W	RESERVED	Reserved	6'b1
[14:9]	R/W	CLKDIV_PCLK	Divide value to create the clock of PCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[8:3]	R/W	CLKDIV_BCLK	Divide value to create the clock of BCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64	6'b1

Bit	R/W	Symbol	Description	Reset Value
			Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	
[2:0]	R/W	CLKSEL_BCLK	Select a clock source. If this bits is N, then PLL[N] is selected as clock source.	3'b0
MDCLK SETTING REGISTER (CLKDIVREG2)				
<i>Address : C001_0028h</i>				
[31:27]	R/W	RESERVED	Reserved	4'b0
[26:21]	R/W	CLKDIV_MPCLK	Divide value to create the clock of MPCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[20:15]	R/W	CLKDIV_MBCLK	Divide value to create the clock of MBCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[14:9]	R/W	CLKDIV_MCLK	Divide value to create the clock of MCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b0
[8:3]	R/W	CLKDIV_MDCLK	Divide value to create the clock of MDCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b0
[2:0]	R/W	CLKSEL_MDCLK	Select a clock source. If this bits is N, then PLL[N] is selected as clock source.	3'b0
GR3DBCLK SETTING REGISTER (CLKDIVREG3)				
<i>Address : C001_002Ch</i>				
[31:27]	R/W	RESERVED	Reserved	4'b0
[26:21]	R/W	RESERVED	Reserved	6'b0
[20:15]	R/W	RESERVED	Reserved	6'b1
[14:9]	R/W	CLKDIV_GR3DPCLK	Divide value to create the clock of GR3DPCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[8:3]	R/W	CLKDIV_GR3DBCLK	Divide value to create the clock of GR3DBCLK. For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[2:0]	R/W	CLKSEL_GR3DBCLK	Select a clock source. If this bits is N, then PLL[N] is selected as clock source.	3'b0
MPEGBCLK SETTING REGISTER (CLKDIVREG4)				
<i>Address : C001_0030h</i>				
[31:27]	R/W	RESERVED	Reserved	4'b0
[26:21]	R/W	RESERVED	Reserved	6'b0

Bit	R/W	Symbol	Description	Reset Value
[20:15]	R/W	RESERVED	Reserved	6'b1
[14:9]	R/W	CLKDIV_MPEGPCLK	Divide value to create the clock of HC MPEGPCLK . For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[8:3]	R/W	CLKDIV_MPEGBCLK	Divide value to create the clock of MPEGBCLK . For 'N' clock divide, enter an [N-1] value. Register Set value 000001 ~ 111111 → actual Divide Value = 2 ~ 64 Ex) For eight clock divide, set this register to 000111b Ex) For three clock divide, set this register to 000010b	6'b1
[2:0]	R/W	CLKSEL_MPEGBCLK	Select a clock source. If this bits is N, then PLL[N] is selected as clock source.	3'b0
RESERVED				
<i>Address : C001_0034h ~ C001_003Fh</i>				
RESERVED				
<i>Address : C001_0040h ~ C001_0047h</i>				
PLL0 SETTING REGISTER FOR SPREAD SPECTRUM (PLLSETREG0_SSCG)				
<i>Address : C001_0048h</i>				
[31 : 16]	R/W	K	Value of 16-bit DSM.	16'b0
[15:8]	R/W	MFR	Modulation frequency control	8'b0
[7:2]	R/W	MRR	Modulation rate control	6'b0
[1:0]	R/W	SEL_PF	Modulation method 00=down spread, 01=up spread, 1x=center spread	2'b0
<Note>This register is applied only for dithered-type PLL.				
PLL1 SETTING REGISTER FOR SPREAD SPECTRUM (PLLSETREG1_SSCG)				
<i>Address : C001_004Ch</i>				
[31 : 16]	R/W	K	Value of 16-bit DSM.	16'b0
[15:8]	R/W	MFR	Modulation frequency control	8'b0
[7:2]	R/W	MRR	Modulation rate control	6'b0
[1:0]	R/W	SEL_PF	Modulation method 00=down spread, 01=up spread, 1x=center spread	2'b0
<Note>This register is applied only for dithered-type PLL.				
PLL2 SETTING REGISTER FOR SPREAD SPECTRUM (PLLSETREG2_SSCG)				
<i>Address : C001_0050h</i>				
[31 : 16]	R/W	K	Value of 16-bit DSM.	16'b0
[15:8]	R/W	MFR	Modulation frequency control	8'b0
[7:2]	R/W	MRR	Modulation rate control	6'b0
[1:0]	R/W	SEL_PF	Modulation method 00=down spread, 01=up spread, 1x=center spread	2'b0
<Note>This register is applied only for dithered-type PLL.				
PLL3 SETTING REGISTER FOR SPREAD SPECTRUM (PLLSETREG3_SSCG)				
<i>Address : C001_0054h</i>				

Bit	R/W	Symbol	Description	Reset Value
[31 : 16]	R/W	K	Value of 16-bit DSM.	16'b0
[15:8]	R/W	MFR	Modulation frequency control	8'b0
[7:2]	R/W	MRR	Modulation rate control	6'b0
[1:0]	R/W	SEL_PF	Modulation method 00=down spread, 01=up spread, 1x=center spread	2'b0

<Note>This register is applied only for dithered-type PLL.

RISING EDGE DETECT ENABLE REGISTER (GPIOWAKEUPRISEENB)

Address : C001_0200h

[31 : 15]	-	RESERVED	Reserved	17b0
[14]		RISEWKSRC14	Wakeup source (USB20OTG.SUSPEND) Rising Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[13]		RISEWKSRC13	Wakeup source (USB20OTG.SLEEP) Rising Edge Detect Enable This bit enables wakeup form power down modes, when USB2.0 OTG's Sleep pin toggles. 0: Disable 1: Enable	1'b0
[12]		RISEWKSRC12	Reserved. This bit must be set to '1'b0'	1'b0
[11]		RISEWKSRC11	Reserved. This bit must be set to '1'b0'	1'b0
[10]		RISEWKSRC10	Wakeup source (UART[3].RX) Rising Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[9]		RISEWKSRC9	Wakeup source (UART[2].RX) Rising Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[8]		RISEWKSRC8	Wakeup source (UART[1].RX) Rising Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[7]		RISEWKSRC7	Wakeup source (UART[0].RX) Rising Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[6]	R/W	RISEWKSRC6	Wakeup source (VDDPWRTOGGLE) Rising Edge Detect Enable This bit enables wakeup form power down modes, when user pushed VDDPWRTOGGLE PAD. 0: Disable 1: Enable	1'b0
[5]	R/W	RISEWKSRC5	Wakeup Source (ALIVEGPIO5) Rising Edge Detect Enable 0 : Disable 1 : Enable	1'b0
[4]	R/W	RISEWKSRC4	Wakeup Source (ALIVEGPIO4) Rising Edge Detect Enable 0 : Disable 1 : Enable	1'b0
[3]	R/W	RISEWKSRC3	Wakeup Source (ALIVEGPIO3) Rising Edge Detect Enable 0 : Disable 1 : Enable	1'b0
[2]	R/W	RISEWKSRC2	Wakeup Source (ALIVEGPIO2) Rising Edge Detect Enable 0 : Disable 1 : Enable	1'b0

Bit	R/W	Symbol	Description	Reset Value
[1]	R/W	RISEWKSRC1	Wakeup Source (ALIVEGPIO1) Rising Edge Detect Enable 0 : Disable 1 : Enable	1'b 1
[0]	R/W	RISEWKSRC0	Wakeup Source (ALIVEGPIO0) Rising Edge Detect Enable 0 : Disable 1 : Enable	1'b 1
FALLING EDGE DETECT ENABLE REGISTER (GPIOWAKEUPFALLENB)				
<i>Address : C001_0204h</i>				
[31:15]	-	RESERVED	Reserved	17'b0
[14]		FALLWKSRC14	Wakeup source (USB20OTG.SUSPEND)Falling Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[13]		FALLWKSRC13	Wakeup source (USB20OTG.SLEEP)Falling Edge Detect Enable This bit enables wakeup form power down modes, when USB2.0 OTG's Sleep pin toggles. 0: Disable 1: Enable	1'b0
[12]		FALLWKSRC12	Reserved. This bit must be set to '1'b0'	1'b0
[11]		FALLWKSRC11	Reserved. This bit must be set to '1'b0'	1'b0
[10]		FALLWKSRC10	Wakeup source (UART[3].RX)Falling Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[9]		FALLWKSRC9	Wakeup source (UART[2].RX)Falling Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[8]		FALLWKSRC8	Wakeup source (UART[1].RX)Falling Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[7]		FALLWKSRC7	Wakeup source (UART[0].RX)Falling Edge Detect Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[6]	R/W	FALLWKSRC6	Wakeup source (VDDPWRTOGGLE) Falling Edge Detect Enable 0: Disable 1: Enable	1'b0
[5]	R/W	FALLWKSRC5	Wakeup Source (ALIVEGPIO5) Falling Edge Detect Enable 0 : Disable 1 : Enable	1'b 0
[4]	R/W	FALLWKSRC4	Wakeup Source (ALIVEGPIO4) Falling Edge Detect Enable 0 : Disable 1 : Enable	1'b 0
[3]	R/W	FALLWKSRC3	Wakeup Source (ALIVEGPIO3) Falling Edge Detect Enable 0 : Disable 1 : Enable	1'b 0
[2]	R/W	FALLWKSRC2	Wakeup Source (ALIVEGPIO2) Falling Edge Detect Enable 0 : Disable 1 : Enable	1'b 0
[1]	R/W	FALLWKSRC1	Wakeup Source (ALIVEGPIO1) Falling Edge Detect Enable 0 : Disable 1 : Enable	1'b 1
[0]	R/W	FALLWKSRC0	Wakeup Source (ALIVEGPIO0) Falling Edge Detect Enable 0 : Disable 1 : Enable	1'b 1
GPIO RESET ENABLE REGISTER (GPIORSTENB)				

Bit	R/W	Symbol	Description	Reset Value
<i>Address : C001_0208h</i>				
[31 : 15]	R/W	RESERVED	Reserved (These bits should always be '0')	17'b0
[14]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[13]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[12]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[11]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[10]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[9]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[8]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[7]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[6]	R/W	RESERVED	Reserved (These bits should always be '0')	1'b0
[5]	R/W	GPIO5RSTENB	ALIVEGPIO5 Reset Source Enable 0 : Disable 1 : Enable	1'b0
[4]	R/W	GPIO4RSTENB	ALIVEGPIO4 Reset Source Enable 0 : Disable 1 : Enable	1'b0
[3]	R/W	GPIO3RSTENB	ALIVEGPIO3 Reset Source Enable 0 : Disable 1 : Enable	1'b0
[2]	R/W	GPIO2RSTENB	ALIVEGPIO2 Reset Source Enable 0 : Disable 1 : Enable	1'b0
[1]	R/W	GPIO1RSTENB	ALIVEGPIO1 Reset Source Enable 0 : Disable 1 : Enable	1'b0
[0]	R/W	GPIO0RSTENB	ALIVEGPIO0 Reset Source Enable 0 : Disable 1 : Enable	1'b0

<Note> Wake-up and Reset Enable are not allowed at the same time for the same ALIVEGPIO.

GPIO WAKEUP ENABLE REGISTER (GPIORSTENB)

Address : C001 020Ch

Bit	R/W	Symbol	Description	Reset Value
			This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	
[8]	R/W	GPIO8WKENB	Wakeup source (UART[1].RX) Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[7]	R/W	GPIO7WKENB	Wakeup source (UART[0].RX) Enable This bit enables wakeup form power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[6]	R/W	VDDTOGGLEWKENB	VDDPWRTOGGLE Wakeup Source Enable 0 : Disable 1 : Enable	1'b0
[5]	R/W	GPIO5WKENB	ALIVEGPIO5 Wakeup Source Enable 0 : Disable 1 : Enable	1'b0
[4]	R/W	GPIO4WKENB	ALIVEGPIO4 Wakeup Source Enable 0 : Disable 1 : Enable	1'b0
[3]	R/W	GPIO3WKENB	ALIVEGPIO3 Wakeup Source Enable 0 : Disable 1 : Enable	1'b0
[2]	R/W	GPIO2WKENB	ALIVEGPIO2 Wakeup Source Enable 0 : Disable 1 : Enable	1'b0
[1]	R/W	GPIO1WKENB	ALIVEGPIO1 Wakeup Source Enable 0 : Disable 1 : Enable	1'b1
[0]	R/W	GPIO0WKENB	ALIVEGPIO0 Wakeup Source Enable 0 : Disable 1 : Enable	1'b1

<Note> Wake-up and Reset Enable are not allowed at the same time for the same ALIVEGPIO.

GPIO INTERRUPT ENABLE REGISTER(INTENB)

Address : C001_0210h

Bit	R/W	Symbol	Description	Reset Value
			0: Disable 1: Enable	
[7]	R/W	GPIO7	Wakeup source (UART[0].RX) Event Interrupt Enable This bit enables wakeup from power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[6]	R/W	VDDTOGGLE	VDDPWRTOGGLE Event Interrupt Enable 0 : Disable 1 : Enable	1'b0
[5]	R/W	GPIO5	ALIVEGPIO5 Event Interrupt Enable 0 : Disable 1 : Enable	1'b0
[4]	R/W	GPIO4	ALIVEGPIO4 Event Interrupt Enable 0 : Disable 1 : Enable	1'b0
[3]	R/W	GPIO3	ALIVEGPIO3 Event Interrupt Enable 0 : Disable 1 : Enable	1'b0
[2]	R/W	GPIO2	ALIVEGPIO2 Event Interrupt Enable 0 : Disable 1 : Enable	1'b0
[1]	R/W	GPIO1	ALIVEGPIO1 Event Interrupt Enable 0 : Disable 1 : Enable	1'b1
[0]	R/W	GPIO0	ALIVEGPIO0 Event Interrupt Enable 0 : Disable 1 : Enable	1'b1

GPIO INTERRUPT PENDING REGISTER (GPIOINTPEND)

Address : C001_0214h

[31:15]	R	RESERVED	Reserved	17'b0
[14]	R/W	GPIO14PEND	This bit is set as '1' when (USB20OTG. SUSPEND) Event occurs. And this bit is cleared by setting as '1' This bit enables wakeup from power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[13]	R/W	GPIO13PEND	This bit is set as '1' when (USB20OTG.SLEEP) Event occurs. And this bit is cleared by setting as '1' This bit enables wakeup from power down modes, when USB2.0 OTG's Sleep pin toggles. 0: Disable 1: Enable	1'b0
[12]	R/W	GPIO12PEND	Reserved. This bit must be set to '1'b0'	1'b0
[11]	R/W	GPIO11PEND	Reserved. This bit must be set to '1'b0'	1'b0
[10]	R/W	GPIO10PEND	This bit is set as '1' when (UART[3].RX) Event occurs. And this bit is cleared by setting as '1' This bit enables wakeup from power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[9]	R/W	GPIO9PEND	This bit is set as '1' when (UART[2].RX) Event occurs. And this bit is cleared by setting as '1' This bit enables wakeup from power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0
[8]	R/W	GPIO8PEND	This bit is set as '1' when (UART[1].RX) Event occurs. And this bit is cleared by setting as '1' This bit enables wakeup from power down modes, when UART RX pin pin toggles. 0: Disable 1: Enable	1'b0

RESET STATUS REGISTER(RESETSTATUS)

Address : C001_0218h

<Note> The priority of Reset – POR > GPIO > Watchdog > Software

INTERRUPT ENABLE REGISTER (INTENABLE)

Address : C001 021Ch

[31 : 2]	-	RESERVED	Reserved	30'b0
[1]	R/W	BATF	BATF(Battery Fault) Event Interrupt Enable Interrupt occurs when BATF is low level. 0 : Disable 1 : Enable	1'b0

Bit	R/W	Symbol	Description	Reset Value
[0]	R/W	RTC	RTC Event Interrupt Enable Interrupt occurs when BATF is low level. 0 : Disable 1 : Enable	1'b0
INTERRUPT PENDING REGISTER (INTPEND)				
<i>Address : C001_0220h</i>				
[31:2]	R-	RESERVED	Reserved	30'b0
[1]	R/W	BATFWAKEUP	This bit is set as '1' when BATF(Battery Fault) Event occurs. And this bit is cleared by setting as '1' Read > 0 : None 1 : Interrupt Pended Write > 0 : Not Clear 1 : Clear	1'b0
[0]	R/W	RTCWAKEUP	This bit is set as '1' when RTC Wakeup Event occurs. And this bit is cleared by setting as '1' Read > 0 : None 1 : Interrupt Pended Write > 0 : Not Clear 1 : Clear	1'b0
POWER MANGEMENT CONTROL REGISTER (PWRCONT)				
<i>Address : C001_0224h</i>				
[31:16]	R	RESERVED	Reserved	16'b0
[15:12]	R/W	USE_WFI	Use STANDBYWFI[n] signal as indicating signal to go into stop mode. .n=0..3	4'b1111
[11:8]	R/W	USE_WFE	Use STANDBYWFE[n] signal as indicating signal to go into stop mode. .n=0..3	4'b1111
[4]	R/W	XTAL_PWRDN	Xystal power down mode selection This controls the power down of Xystal-PAD in stop-mode. 1=XTAL is not powered down in stop mode. 0=XTAL is powered down instop mode.	1'b0
[3]	R/W	SWRSTENB	Software Reset Enable. 0 : Disable 1 : Enable	1'b0
[2]	R/W	RESERVED	Reserved (This bit always should be '0')	1'b0
[1]	R/W	RTCWKENB	RTC Wake-up enable 0 : Disable 1 : Enable	1'b0
[0]	R/W	RESERVED	Reserved (This bit always should be '0')	1'b0
POWER MANGEMENT MODE REGISTER (PWRMODE)				
<i>Address : C001_0228h</i>				
[31:16]	R/W	RESERVED	Reserved	16'b0
[15]	R/W	CHGPLL	Change PLL Value with new value defined in PLL Setting Register (PLL0set, PLL1set) in clock Controller. Read > 0 : Stable 1 : PLL is Unstable Write > 0 : None 1 : PLL Value Change	1'b0
[14:13]		RESERVED	Reserved	2'b0
[12]	W	SWRST	This bit is cleared after Software Reset 0 : Do Not Reset 1 : go to Reset	1'b0
[11:6]	R	RESERVED	Reserved	6'b0
[5]	R	LASTPWRSTOP	Indicates that the chip has been in STOP Mode before in Normal state.(This bit is cleared in case of Reset in Normal state) 0 : None 1 : Stop mode	1'b0

Bit	R/W	Symbol	Description	Reset Value
[11]	R	CFGSTBUSWIDTH	Static Memory BUS bit. Internal ROM boot \wedge fixed 0. 0 : 8bit 1 : 16 bit	DISD3
[10]	R	CFGALTERNATIVE	Indicates eMMC Alternative Boot mode when (CfgBootMode == 4'bx101). 0: Alternative Boot 1: Normal Boot Indicates [1] bit of Serial flash memory address when (CfgBootMode == 4'bx100) Indicates UART Baudrate when (CfgBootMode == 4'bx011) 0: 19200bps 1: 115200bps Indicates Serial flash memory address width when (CfgBootMode == 4'bx100). (CfgAlternative, CfgPARTITION) == 2'b00 : 16bit address (CfgAlternative, CfgPARTITION) == 2'b01 : 24bit address (CfgAlternative, CfgPARTITION) == 2'b10 : 25bit address	DISD2
[9]	R	CFGPARTITION	Indicates Boot Partition on eMMC when (CfgBootMode == 4'bx101) 0: Default Partition 1: Boot Partition (Partition#1) Used for [0] bit of Serial flash memory address when (CfgBootMode == 4'bx100).	DISD1
[8]	R	CFGLATADDR	Static Memory Latched Address. 0 : None 1 : Latched	DISD0
[7]	R	CFGEMMCBOOTMODE	Indicates eMMC Boot mode when (CfgBootMode == 4'bx101). 0: Normal SD Boot 1: eMMC Boot	SD7
[6]	R	CFGOTGSESSIONCHECK	Indicates USB OTG Session Check when (CfgBootMode == 4'bx110). 0: not check 1: check	SD6
[5]	R	CFGICACHE	Indicates L1 Cache enable when (CfgBootMode != F). Used for CPU Instruction cache enable in the case of internal rom boot. 0 : disable 1 : enable	SD5
[4]	R	CFGDECRYPT	Indicates AES ECB mode decrypt when (CfgBootMode != F) 0: not decrypt 1: decrypt	SD4
[3]	R	CFGNANDSELCS	Select nand chip 0 or 1 when (CfgBootMode == 7). 0: nNCS[0] 1 : nNCS[1] In the case of (CfgBootMode == 4'hF), CfgNANDSELCS is ignored and nNCS[1] can be used only	SD3
[2]	R	CFGNANDPAGE	Indicates External nand page size when (CfgBootMode == 7). 0: 2K or below 1: 4K or above	SD2
[1:0]	R	CFGNANDTYPE	Indicates External nand type when (CfgBootMode == 7). 0: Small Block 3 Address 1: Small block 4 Address 2: Large 4 Address 3: Large 5 Address	2'b{SD1, SD0}

Bit	R/W	Symbol	Description	Reset Value
RESERVED				
Address : C001_0240h ~ C001_03FFh				

Section 5. Clock Generator

5.1 IP Clock Generator Overview

The IP Clock Generator can generate divided clock. Each IP have clocking scheme which requires several different division ratio simultaneously. Therefore, each of IP Clock Generator supplies required clock to each IP. These IP Clock Generators uses the PLL from SYSCTRL or External Clock from PAD. And it can divide required clock of each IP by 2^n divider.

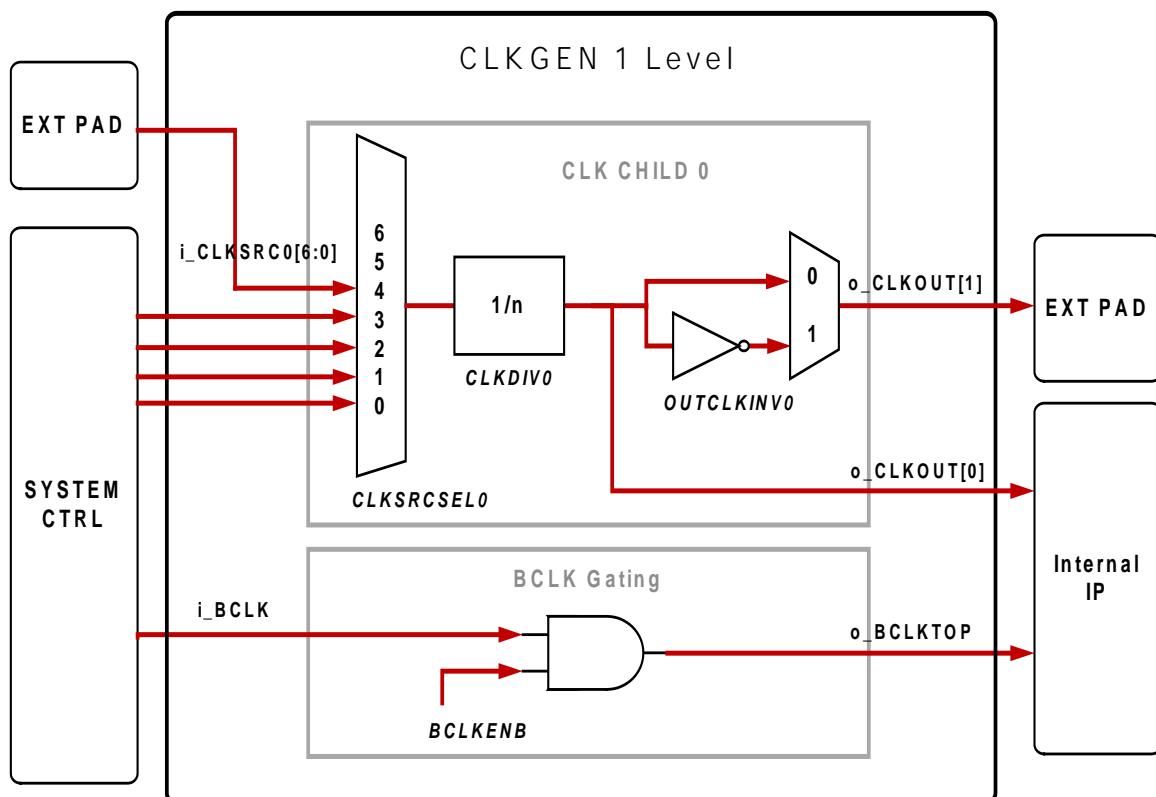


Figure 5-1. Interconnection Example of Clock Generator

5.1.1 Clock Generator Level 0

Clock Generator Level 0 does not have clock divider. It can only do clock gating. It uses PCLK or BCLK Gating.

- Following peripherals use Level 0 Clock Generator:

- CODA960
- Crypto
- I2C
- Mali400
- MPEGTSI
- PDM
- SCALLER
- DEINTERLACE
- MLC

5.1.1.1 Block Diagram

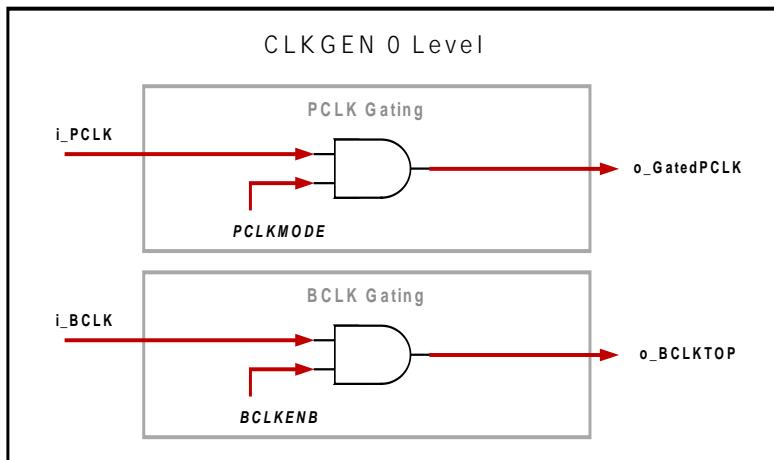


Figure 5-2. Block Diagram of Clock Generator Level 0

5.1.1.2 Register Summary

Bit	R/W	Symbol	Description	Reset Value
CODA960				
Clock Generator Enable Register (CODA960CLKENB)				
Address : 0xC00C7000				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0: PCLK is only enabled when CPU accesses this module 1: PCLK is always enabled	1'b0
[2:0]	R	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	2'b0
Crypto				
Clock Generator Enable Register (CRYPTOCLKENB)				
Address : 0xC00C6000				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0: PCLK is only enabled when CPU accesses this module 1: PCLK is always enabled	1'b0
[2:0]	R	RESERVED	Reserved	2'b0
I2C				
Clock Generator Enable Register (I2CCLKENB)				
Address : 0xC00AE000 / 0xC00AF000 / 0xC00B0000				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0: PCLK is only enabled when CPU accesses this module 1: PCLK is always enabled	1'b0
[3:0]	R	RESERVED	Reserved	3'b0
Mali400				

Bit	R/W	Symbol	Description	Reset Value
Clock Generator Enable Register (MALI400CLKENB)				
Address : 0xC00C3000				
[31:2]	R	RESERVED	Reserved	30'b0
[1:0]	R	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	2b0
MPEGTSI				
Clock Generator Enable Register (MPEGTSICLKENB)				
Address : 0xC00CB700				
[31:2]	R	RESERVED	Reserved	30'b0
[1:0]	R	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	2b0
PDM				
Clock Generator Enable Register (PDMCLKENB)				
Address : 0xC00CB000				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0 : PCLK is only enabled when CPU accesses this module 1 : PCLK is always enabled	1b0
[2:0]	R	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	3b0
SCALER				
Clock Generator Enable Register (SCALERCLKENB)				
Address : 0xC00B6000				
[31:2]	R	RESERVED	Reserved	30'b0
[1:0]	R	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	2b0
DEINTERLACE				
Clock Generator Enable Register (DEINTERLACECLKENB)				
Address : 0xC00B5000				
[31:2]	R	RESERVED	Reserved	30'b0
[1:0]	R	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	2b0
MLC				
Clock Generator Enable Register (MLCCLKENB)				
Address : 0xC01023C0 / 0xC01027C0				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0 : PCLK is only enabled when CPU accesses this module 1 : PCLK is always enabled	1b0
[2]	R	RESERVED	Reserved	1b0
[1:0]	R	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	2b0

5.1.2 Clock Generator Level 1

The Clock Generator Level-1 has one clock divider. Clock Divider has 8-bit divide registers. Divide registers can reach to 256 levels and it can divide up to 256 levels.

- Following peripherals use Level 1 Clock Generator:

- MIPCSI
- PPM
- PWMTIMER
- SDMMC
- SPDIFTX
- SSP
- UART
- VIP

5.1.2.1 Block Diagram

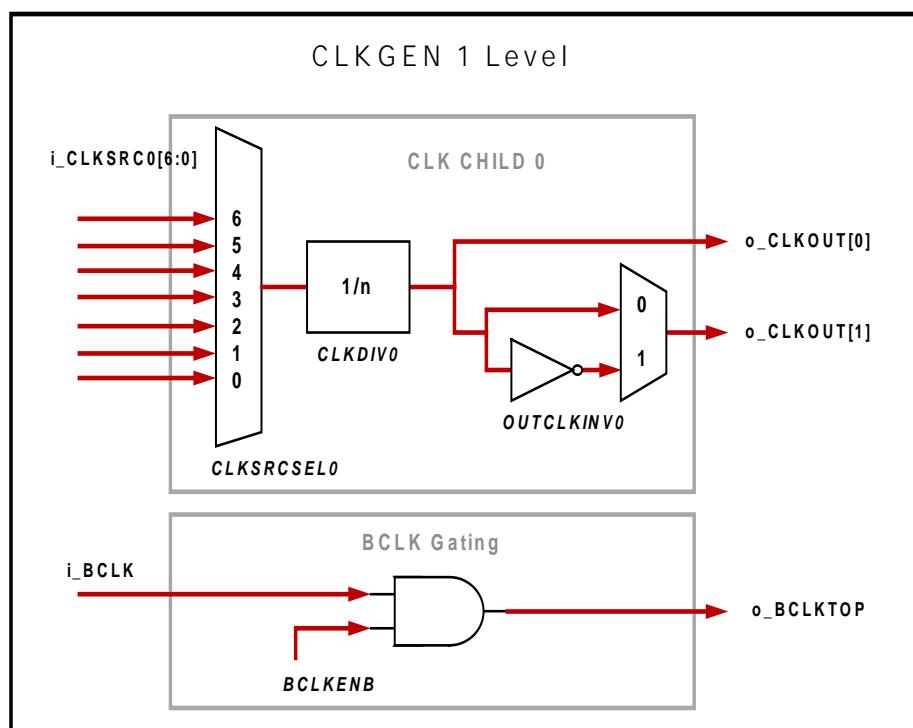


Figure 5-3. Block Diagram of Clock Generator Level 1

5.1.2.2 Register Summary

Bit	R/W	Symbol	Description	Reset Value
MIPCSI				
Clock Generator Enable Register (MIPCSICLKENB)				
Address: 0xC00CA000				
[31:3]	R	RESERVED	Reserved	29'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock.	1b0

Bit	R/W	Symbol	Description	Reset Value
			0 : Disable 1 : Enable	
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (MIPICSICLKGEN0L)				
Address: 0xC00CA004				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2]	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0
PPM				
Clock Generator Enable Register (PPMCLKENB)				
Address: 0xC00C4000				
[31:3]	R	RESERVED	Reserved	29b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (PPMCLKGEN0L)				
Address: 0xC00C4004				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2]	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0
PWMTIMER				
Clock Generator Enable Register (PWMTIMERCLKENB)				
Address: 0xC00BA000 / 0xC00BE000 / 0xC00BF000 / 0xC00C0000 (PWM)				
Address: 0xC00B9000 / 0xC00BB000 / 0xC00BC000 / 0xC00BD000 (TIMER)				
[31:3]	R	RESERVED	Reserved	29b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (PWMTIMERCLKGEN0L)				

Bit	R/W	Symbol	Description	Reset Value
Address: 0xC00BA004 / 0xC00BE004 / 0xC00BF004 / 0xC00C0004 (PWM)				
Address: 0xC00B9004 / 0xC00BB004 / 0xC00BC004 / 0xC00BD004 (TIMER)				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1'b0
[14:13]	R	RESERVED	Reserved	2'b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2]	3'b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
SDMMC				
Clock Generator Enable Register (SDMMCCLKENB)				
Address: 0xC00C5000 / 0xC00CC000 / 0xC00CD000				
[31:3]	R	RESERVED	Reserved	29'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1'b0
[1:0]	R	RESERVED	Reserved	2'b0
Clock Generator Control 0 Low Register (SDMMCCLKGEN0L)				
Address: 0xC00C5004 / 0xC00CC004 / 0xC00CD004				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1'b0
[14:13]	R	RESERVED	Reserved	2'b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2]	3'b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
SPDIFTX				
Clock Generator Enable Register (SPDIFTXCLKENB)				
BASE Address: 0xC00B8000				
[31:3]	R	RESERVED	Reserved	29'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1'b0
[1:0]	R	RESERVED	Reserved	2'b0
Clock Generator Control 0 Low Register (SPDIFTXCLKGEN0L)				
BASE Address: 0xC00B8004				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when	1'b0

Bit	R/W	Symbol	Description	Reset Value
			CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2]	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0
SSP				
Clock Generator Enable Register (SSPCLKENB)				
Address: 0xC00AC000 / 0xC00AD000 / 0xC00A7000				
[31:3]	R	RESERVED	Reserved	29b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (SSPCLKGEN0L)				
Address: 0xC00AC004 / 0xC00AD004 / 0xC00A7004				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2]	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0
UART				
Clock Generator Enable Register (UARTCLKENB)				
Address: 0xC00A9000 / 0xC00AA000 / 0xC00A8000 / 0xC00AB000 / 0xC006E000 / 0xC0084000				
[31:3]	R	RESERVED	Reserved	29b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (UARTCLKGEN0L)				
Address: 0xC00A9004 / 0xC00AA004 / 0xC00A8004 / 0xC00AB004 / 0xC006E004 / 0xC0084004				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock.	8b0

Bit	R/W	Symbol	Description	Reset Value
			Divider value = CLKDIV0 + 1	
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2]	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0
VIP				
Clock Generator Enable Register (VIPCLKENB)				
Address: 0xC00C1000 / 0xC00C2000				
[31:3]	R	RESERVED	Reserved	29b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R/W	BCLKENB	BCLK Enable 0: Disable 1: Reserved 2: Reserved 3: Always	2b0
Clock Generator Control 0 Low Register (VIPCLKGEN0L)				
Address: 0xC00C1004 / 0xC00C2004				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: CIS External Clock 0 5: CIS External Clock 1	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0

5.1.3 Clock Generator Level 2

The Clock Generator Level 2 has two clock dividers. Clock Divider has 8-bit divide registers. And each divide registers can reach to 256 levels and it can divide up to 256 levels. The two clock divider is serialized. Therefore Clock Generator Level-2 can divide up to 65,536.

- Following peripherals use Level 2 Clock Generator:
 - GMAC
 - I2S
 - USBHOSTOTG
 - DPC
 - LVDS
 - HDMI
 - MIPIDSI

5.1.3.1 Block Diagram

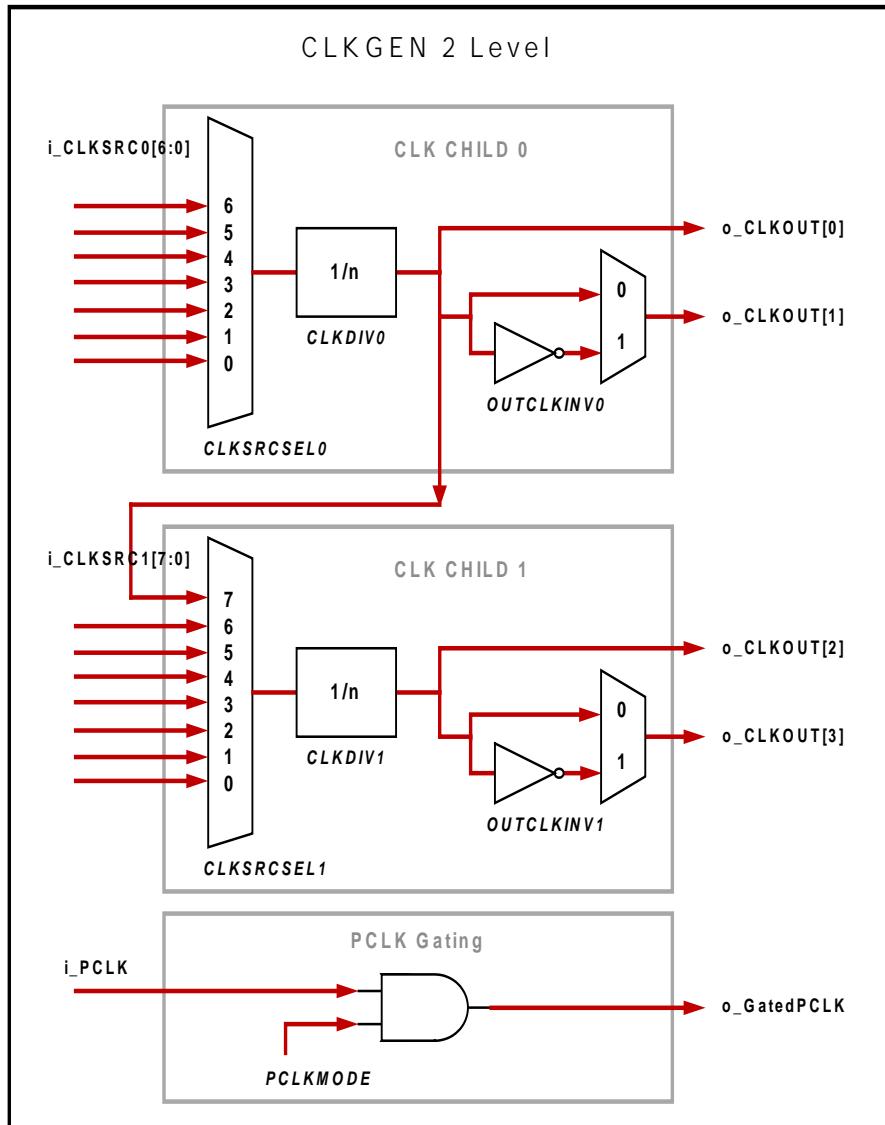


Figure 5-4. Block Diagram of Clock Generator Level 2

5.1.3.2 Register Summary

Bit	R/W	Symbol	Description	Reset Value
GMAC				
Clock Generator Enable Register (GMACCLKENB)				
Address 0xC00C8000				
[31:3]	R	RESERVED	Reserved	29'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (GMACCLKGEN0L)				
Address 0xC00C8004				

Bit	R/W	Symbol	Description	Reset Value
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1'b0
[14:13]	R	RESERVED	Reserved	2'b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: External RX Clock	3'b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
Clock Generator Reserved Area				
Address 0xC00C8008				
[31:0]	R	RESERVED	Reserved	
Clock Generator Control 1 low Register (GMACCLKGEN1L)				
Address 0xC00C800C				
[31:13]	R	RESERVED	Reserved	19'b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL1	0: Reserved 1: Reserved 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Divided Clock from Divider 0	3'b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0 : Bypass 1 : source clock /2 ns	1'b0
I2S				
Clock Generator Enable Register (I2SCLKENB)				
Address 0xC00B2000 / 0xC00B3000 / 0xC00B4000				
[31:3]	R	RESERVED	Reserved	29'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1'b0
[1:0]	R	RESERVED	Reserved	2'b0
Clock Generator Control 0 Low Register (I2SCLKGEN0L)				
Address 0xC00B2004 / 0xC00B3004 / 0xC00B4004				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1'b0
[14:13]	R	RESERVED	Reserved	2'b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: External I2S Codec Clock 2	3'b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0

Bit	R/W	Symbol	Description	Reset Value
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
Clock Generator Reserved Area				
Address 0xC00B2008 / 0xC00B3008 / 0xC00B4008				
[31:0]	R	RESERVED	Reserved	
Clock Generator Control 1 low Register (I2SCLKGEN1L)				
Address 0xC00B200C / 0xC00B300C / 0xC00B400C				
[31:13]	R	RESERVED	Reserved	19'b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL1	0: Reserved 1: Reserved 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Divided Clock from Divider 0	3'b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0 : Bypass 1 : source clock /2 ns	1'b0
USBHOSTOTG				
Clock Generator Enable Register (USBHOSTOTGCLKENB)				
Address 0xC006B000				
[31:3]	R	RESERVED	Reserved	29'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1'b0
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (USBHOSTOTGCLKGEN0L)				
Address 0xC006B004				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3]	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
Clock Generator Reserved Area				
Address 0xC006B008				
[31:0]	R	RESERVED	Reserved	
Clock Generator Control 1 low Register (USBHOSTOTGCLKGEN1L)				
Address 0xC006B00C				
[31:13]	R	RESERVED	Reserved	19'b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0

Bit	R/W	Symbol	Description	Reset Value
[4:2]	R/W	CLKSRCSEL1	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: External XT1 5: Reserved 6: Reserved 7: Divided Clock from Divider 0	3b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0; Bypass 1 source clock /2 ns	1b0
DPC				
Clock Generator Enable Register (DPCCCLKENB)				
Address: 0xC0102BC0 / 0xC0102BC0				
[31:4]	R	RESERVED	Reserved	28b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0: PCLK is only enabled when CPU accesses this module 1 : PCLK is always enabled	1b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R	RESERVED	Reserved	2b0
Clock Generator Control 0 Low Register (DPCCCLKGEN0L)				
Address: 0xC0102BC4 / 0xC0102FC4				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: Reserved 4: HDMI PLL Clock 5: Reserved 6: PLL[3]	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0
Clock Generator Reserved Area				
Address: 0xC0102BC8 / 0xC0102FC8				
[31:0]	R	RESERVED	Reserved	
CLOCK GENERATOR CONTROL 1 LOW REGISTER (DPCCCLKGEN1L)				
ADDRESS: 0XC0102BCCC / 0XC0102FCC				
[31:13]	R	RESERVED	Reserved	19b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL1	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: Reserved 4: HDMI PLL Clock 5: Reserved 6: PLL[3] 7: Divided Clock from Divider 0	3b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0; Bypass 1 source clock /2 ns	1b0

Bit	R/W	Symbol	Description	Reset Value
LVDS				
Clock Generator Enable Register (LVDSCLKENB)				
Address: C0108000				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0 : PCLK is only enabled when CPU accesses this module 1 : PCLK is always enabled	1'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1'b0
[1:0]	R	RESERVED	Reserved	2'b0
Clock Generator Control 0 Low Register (LVDSCLKGEN0L)				
Address: C0108004				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0[1] is 3 or 4. 0 : Enable(Output) 1 : Reserved	1'b0
[14:13]	R	RESERVED	Reserved	2'b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: HDMI PLL Clock 5: Reserved 6: Reserved	3'b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
Clock Generator Reserved Area				
Address: C0108008				
[31:0]	R	RESERVED	Reserved	
Clock Generator Control 1 low Register (LVDSCLKGEN1L)				
Address: C010800C				
[31:13]	R	RESERVED	Reserved	19'b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL1	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: HDMI PLL Clock 5: Reserved 6: Reserved 7: Divided Clock from Divider 0	3'b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0: Bypass 1 : source clock /2 ns	1'b0
HDMI				
Clock Generator Enable Register (HDMICLKENB)				
Address: C0109000				
[31:4]	R	RESERVED	Reserved	28'b0

Bit	R/W	Symbol	Description	Reset Value
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0 : PCLK is only enabled when CPU accesses this module 1 : PCLK is always enabled	1'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1'b0
[1:0]	R	RESERVED	Reserved	2'b0
Clock Generator Control 0 Low Register (HDMICLKGEN0L)				
Address: C0109004				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1'b0
[14:13]	R	RESERVED	Reserved	2'b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: Reserved 5: Reserved 6: Reserved	3'b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
Clock Generator Reserved Area				
Address: C0109008				
[31:0]	R	RESERVED	Reserved	
Clock Generator Control 1 low Register (HDMICLKGEN1L)				
Address: C010900C				
[31:13]	R	RESERVED	Reserved	19'b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL1	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: Reserved 5: Reserved 6: Reserved 7: Divided Clock from Divider 0	3'b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0: Bypass 1 : source clock /2 ns	1'b0
MIPDSI				
Clock Generator Enable Register (MIPDSICLKGENB)				
Address: C0105000				
[31:3]	R	RESERVED	Reserved	29'b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1'b0
[1:0]	R	RESERVED	Reserved	2'b0
Clock Generator Control 0 Low Register (MIPDSICLKGEN0L)				
Address: C0105004				
[31:16]	R	RESERVED	Reserved	16'b0

Bit	R/W	Symbol	Description	Reset Value
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1'b0
[14:13]	R	RESERVED	Reserved	2'b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL0	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: HDMI PLL Clock 5: Reserved 6: Reserved	3'b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1'b0
Clock Generator Reserved Area				
Address: C0105008				
[31: 0]	R	RESERVED	Reserved	
Clock Generator Control 1 low Register (MIPDSICLKGGEN1L)				
Address: C010500C				
[31:13]	R	RESERVED	Reserved	19'b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8'b0
[4:2]	R/W	CLKSRCSEL1	0: PLL[0] 1: PLL[1] 2: PLL[2] 3: PLL[3] 4: HDMI PLL Clock 5: Reserved 6: Reserved 7: Divided Clock from Divider 0	3'b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1'b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0 : Bypass 1 : source clock /2 ns	1'b0

Section 6. System L2 Cache (PL-310 L2C)

6.1 Overview

The addition of an on-chip secondary cache, also referred to as a Level 2 or L2 cache, is a recognized method of improving the performance of ARM-based systems when significant memory traffic is generated by the processor. By definition a secondary cache assumes the presence of a Level 1 or primary cache, closely coupled or internal to the processor.

Memory access is fastest to L1 cache, followed closely by L2 cache. Memory access is typically significantly slower with L3 main memory.

6.1.1 Features

The cache controller features:

- Slave and master AMBA AXI interfaces designed for high performance systems.
- Lockdown format C supported, for data and instructions.
- Lockdown by line supported.
- Lockdown by master ID supported.
- L2 cache available size can be 16KB to 8MB, depending on configuration and the use of the lockdown registers.
- Fixed line length of 32 bytes, eight words or 256 bits.
- Interface to data RAM is byte writable.
- Supports all of the AXI cache modes:
 - write-through and write-back
 - read allocate, write allocate, read and write allocate.
- Force write allocate option to always have cacheable writes allocated to L2 cache, for processors not supporting this mode.
- Normal memory non-cacheable shared reads are treated as cacheable non-allocatable. Normal memory non-cacheable shared writes are treated as cacheable write-through no write-allocate. There is an option, Shared Override, to override this behavior.
- Critical word first linefill supported.
- Pseudo-Random, or round-robin victim selection policy. You can make this deterministic with use of lockdown registers.
- Four 256-bit Line Fill Buffers (LFBs), shared by the master ports. These buffers capture linefill data from main memory, waiting for a complete line before writing to L2 cache memory.
- Two 256-bit Line Read Buffers (LRBs) for each slave port. These buffers hold a line from the L2 memory for a cache hit.
- Three 256-bit Eviction Buffers (EBs). These buffers hold evicted lines from the L2 cache, to be written back to main memory.
- Three 256-bit Store Buffers (STBs). These buffers hold bufferable writes before their draining to main memory, or the L2 cache. They enable multiple writes to the same line to be merged.
- Software option to enable exclusive cache configuration.

- Configuration registers accessible using address decoding in the slave ports.
- Address filtering in the master ports enabling redirection of a certain address range to one master port while all other addresses are redirected to the other one.

6.1.2 Block Diagram

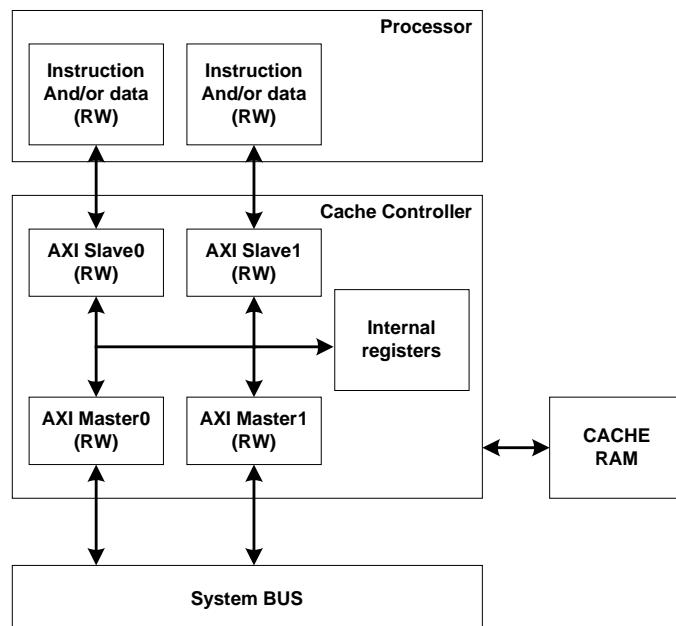


Figure 6-1. System L2 Cache Block Diagram

6.2 Functional Description

6.2.1 L2 Cache User Config

- System L2 Cache Base Address: Base Address: 0xCF000000
- Turn off and Turn on L2C:
 - Turn off: offset 0x100 (Base Address + 0x100) set to 0
 - Turn on: offset 0x100 (Base Address + 0x100) set to 1
- Early Write Response:

The AXI protocol specifies that the write response can only be sent back to an AXI master when the last write data has been accepted. This optimization enables the L2C-310 to send the write response of certain write transactions as soon as the store buffer accepts the write address. This behavior is not compatible with the AXI protocol and is disabled by default. You enable this optimization by setting the Early BRESP Enable bit in the Auxiliary Control Register, bit[30]. The L2C-310 slave ports then send an early write response only if the input signal AWUSERSx[11], x=0 or 1, is set to 1'b1 for the corresponding write transaction.

6.2.2 Initialization Sequence

As an example, a typical cache controller start-up programming sequence consists of the following register operations:

- 1) Write to the Auxiliary, Tag RAM Latency, Data RAM Latency, Prefetch, and Power Control registers using a read-modify-write to set up global configurations:
 - associativity, Way Size
 - latencies for RAM accesses
 - allocation policy
 - prefetch and power capabilities.
- 2) Secure write to the Invalidate by Way, offset 0x77C, to invalidate all entries in cache:
 - Write 0xFFFF to 0x77C
 - Poll cache maintenance register until invalidate operation is complete.
- 3) Write to the Lockdown D and Lockdown I Register 9 if required.
- 4) Write to interrupt clear register to clear any residual raw interrupts set.
- 5) Write to the Interrupt Mask Register if you want to enable interrupts.
- 6) Write to Control Register 1 with the LSB set to 1 to enable the cache.

If you write to the Auxiliary, Tag RAM Latency, or Data RAM Latency Control Register with the L2 cache enabled, this results in a SLVERR. You must disable the L2 cache by writing to the Control Register 1 before Writing to the Auxiliary, Tag RAM Latency, or Data RAM Latency Control Register.

6.3 Register Summary

The base address of the Cache controller is 0xCF000000.

Bit	R/W	Symbol	Description	Reset Value		
Cache ID and type Register						
<i>Address : CF00_0000h</i>						
[31:24]	R	IMPLEMENTER	ARM	0x4100C4C8		
[23:16]		RESERVED	Reserved			
[15:10]		CACHE ID	Cache Controller ID			
[9:6]		PARTNUM	Part number 0x3 denotes CoreLink Level 2 Cache Controller L2C-310			
[5:0]		RTL RELEASE	RTL release 0x8 denotes r3p2 code of the cache controller.			
CACHE Type register						
<i>Address : CF00_0004h</i>						
[31]	R	DATA BANK	0: Data banking not implemented 1: Data banking implemented.	0x1A340340		
[30:29]		RESERVED	Reserved			
[28:25]		CTYPE	4'b11xy x=1 if pl310_LOCKDOWN_BY_MASTER is defined, otherwise 0 y=1 if pl310_LOCKDOWN_BY_LINE is defined, otherwise 0.			
[24]		H	0: unified 1: Harvard			
[23:19]		DSIZE	[23] SBZ/RAZ [22:20] Read from Auxiliary Control Register[19:17] [19] SBZ/RAZ			
[18]		L2 ASSOCIATIVITY	Read from Auxiliary Control Register[16]			
[17:14]		RESERVED	Reserved			
[13:12]		L2 CACHE LINE LENGTH	00-32bytes			
[11:7]		ISIZE	[11] SBZ/RAZ [10:8] Read from Auxiliary Control Register[19:17] [7] SBZ/RAZ			
[6]		L2 ASSOCIATIVITY	Read from Auxiliary Control Register[16]			
[5:2]		RESERVED	Reserved			
[1:0]		L2 CACHE LINE LENGTH	00-32 bytes			
control register						
<i>Address : CF00_0100h</i>						
[31:1]	R	RESERVED	Reserved	0x00000000		
[0]	R/W	ENB	0: L2 Cache disabled. Default 1: L2 Cache Enabled.			
aux control register						
<i>Address : CF00_0104h</i>						

Bit	R/W	Symbol	Description	Reset Value
[31]	R/W	RESERVED	Reserved	0x02070000b
[30]		EARLY BRESP	0 Early BRESP disabled. This is the default. 1 Early BRESP enabled.	
29		INSTRUCTION PREFETCH ENABLE	0 Instruction prefetching disabled. This is the default. 1 Instruction prefetching enabled.	
28		DATA PREFETCH ENABLE	0 Data prefetching disabled. This is the default. 1 Data prefetching enabled.	
27		NS INTERRUPT ACCESS CONTROL	0 Interrupt Clear, 0x220, and Interrupt Mask, 0x214, can only be modified or read with secure accesses. This is the default. 1 Interrupt Clear, 0x220, and Interrupt Mask, 0x214, can be modified or read with secure or non-secure accesses.	
26		NS LOCKDOWN	0 Lockdown registers cannot be modified using non-secure accesses. This is the default. 1 Non-secure accesses can write to the lockdown registers.	
25		CACHE REPLACEMENT POLICY	0 Pseudo-random replacement using lfsr. 1 Round-robin replacement. This is the default.	
24:23		FORCE WRITE ALLOCATE	0b00 Use AWCACHE attributes for WA. This is the default. 0b01 Force no allocate, set WA bit always 0. 0b10 Override AWCACHE attributes, set WA bit always 1, all cacheable write misses become write allocated. 0b11 Internally mapped to 00.	
22		SHARED ATTRIBUTE OVERRIDE ENABLE	0 Treats shared accesses. This is the default. 1 Shared attribute internally ignored.	
21		PARITY ENABLE	0 Disabled. This is the default. 1 Enabled.	
20		EVENT MONITOR BUS ENABLE	0 Disabled. This is the default. 1 Enabled.	
19:17		WAY-SIZE	0b000 Reserved, internally mapped to 16KB. 0b001 16KB 0b010 32KB 0b011 64KB 0b100 128KB 0b101 256KB 0b110 512KB 0b111 Reserved, internally mapped to 512 KB.	
16		ASSOCIATIVITY	0 8-way 1 16-way.	
15:14		RESERVED	SBZ/RAZ	
13		SHARED ATTRIBUTE INVALIDATE ENABLE	0 Shared invalidate behavior disabled. This is the default. 1 Shared invalidate behavior enabled, if Shared Attribute Override Enable bit not set.	
12		EXCLUSIVE CACHE CONFIGURATION	0 Disabled. This is the default. 1 Enabled.	
11		STORE BUFFER DEVICE LIMITATION ENABLE	0 Store buffer device limitation disabled. Device writes can take all slots in store buffer. This is the default.	

Bit	R/W	Symbol	Description	Reset Value
	R/W		1 Store buffer device limitation enabled. Device writes cannot take all slots in store buffer when connected to the Cortex-A9 MPCore processor. There is always one available slot to service Normal Memory. HIGH PRIORITY FOR SO AND DEV READS ENABLE 0 Strongly Ordered and Device reads have lower priority than cacheable accesses when arbitrated in the L2CC L2C-310 master ports. This is the default. 1 Strongly Ordered and Device reads get the highest priority when arbitrated in the L2C-310 master ports.	0x00000777
10		RESERVED	SBZ/RAZ	
9:1		FULL LINE OF ZERO ENABLE	0 Full line of write zero behavior disabled. This is the default. 1 Full line of write zero behavior Enabled.	
0			tag and data ram control register Address : CF00_0108h ~ CF00_010Ch	
[31:11]	R/W	RESERVED	Reserved	0x00000777
[10:8]		RAM WRITE ACCESS LATENCY	Default value depends on the value of pl310_TAG_WRITE_LAT for reg1_tag_ram_control or pl310_DATA_WRITE_LAT for reg1_data_ram_control. 0b000 1 cycle of latency, there is no additional latency 0b001 2 cycles of latency 0b010 3 cycles of latency 0b011 4 cycles of latency 0b100 5 cycles of latency 0b101 6 cycles of latency 0b110 7 cycles of latency 0b111 8 cycles of latency.	
[7]		RESERVED	Reserved	
[6:4]		RAM READ ACCESS LATENCY	0b000 1 cycle of latency, there is no additional latency 0b001 2 cycles of latency 0b010 3 cycles of latency 0b011 4 cycles of latency 0b100 5 cycles of latency 0b101 6 cycles of latency 0b110 7 cycles of latency 0b111 8 cycles of latency.	
[3]		RESERVED	SBZ/RAZ	
[2:0]		RAM SETUP LATENCY	0b000 1 cycle of latency, there is no additional latency 0b001 2 cycles of latency 0b010 3 cycles of latency 0b011 4 cycles of latency 0b100 5 cycles of latency 0b101 6 cycles of latency 0b110 7 cycles of latency 0b111 8 cycles of latency.	
event counter control register Address : CF00_0200h				
[31:3]	R/W	RESERVED	SBZ/RAZ	0x00000000

Bit	R/W	Symbol	Description	Reset Value		
[2:1]	R/W	COUNTER RESET	Always Read as zero. The following counters are reset when a 1 is written to the following bits: bit[2] = Event Counter1 reset bit[1] = Event Counter0 reset.			
[0]		EVENT COUNTER ENABLE	0 Event Counting Disable. This is the default. 1 Event Counting Enable.			
event counter configuration register 1, 0						
<i>Address : CF00_0204h ~ CF00_0208h</i>						
[31:6]	R/W	RESERVED	Reserved	0x00000000		
[5:2]		COUNTER EVENT SOURCE	Event Encoding Counter Disabled: 0b'0000 CO: 0b'0001 DRHIT: 0b'0010 DRREQ: 0b'0011 DWHIT: 0b'0100 DWREQ: 0b'0101 DWTREQ: 0b'0110 IRHIT: 0b'0111 IRREQ: 0b'1000 WA: 0b'1001 IPFALLOC: 0b'1010 EPFHIT: 0b'1011 EPFALLOC: 0b'1100 SRRCVD: 0b'1101 SRCONF: 0b'1110 EPFRCVD: 0b'1111			
[1:0]		EVENT COUNTER INTERRUPT GENERATION	0b00 Disabled. This is the default. 0b01 Enabled: Increment condition. 0b10 Enabled: Overflow condition. 0b11 Interrupt generation is disabled.			
event counter registers 1, 0						
<i>Address : CF00_020ch ~ CF00_0210h</i>						
[31:0]	R/W	COUNTER	Total of the event selected. If a counter reaches its maximum value, it saturates at that value until it is reset.	0x00000000		
Interrupt mask register						
<i>Address : CF00_0214h</i>						
[31:9]	R/W	RESERVED	Reserved	0x00000000		
[8]		DECERR	DECERR from L3 0: Masked, Default 1: Enabled.			
[7]		SLVERR	SLVERR from L3 0: Masked, Default 1: Enabled.			
[6]		ERRRD	Error on L2 data RAM, Read 0: Masked, Default 1: Enabled.			
[5]		ERRRT	Error on L2 tag RAM, Read			

Bit	R/W	Symbol	Description	Reset Value
			0: Masked, Default 1: Enabled.	
[4]		ERRWD	Error on L2 data RAM, Write 0: Masked, Default 1: Enabled.	
[3]		ERRWT	Error on L2 tag RAM, Write 0: Masked, Default 1: Enabled.	
[2]		PARRD	Parity Error on L2 data RAM, Read 0: Masked, Default 1: Enabled.	
[1]		PARRT	Parity Error on L2 tag RAM, Read 0: Masked, Default 1: Enabled.	
[0]		ECNTR	Even Counter1 and Event Counter 0 Overflow Increment 0: Masked, Default 1: Enabled.	
Interrupt mask STATUS register				
<i>Address : CF00_0218h</i>				
[31:9]		RESERVED	Reserved	
[8]		DECERR	DECERR from L3 HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	
[7]		SLVERR	SLVERR from L3 HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	
[6]		ERRRD	Error on L2 data RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	
[5]		ERRRT	Error on L2 tag RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	
[4]	R	ERRWD	Error on L2 data RAM, Write HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	0x00000000
[3]		ERRWT	Error on L2 tag RAM, Write HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	
[2]		PARRD	Parity Error on L2 data RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	
[1]		PARRT	Parity Error on L2 tag RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.	
[0]		ECNTR	Even Counter1 and Event Counter 0 Overflow Increment	

Bit	R/W	Symbol	Description	Reset Value		
			HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, either no interrupt has been generated, or the interrupt is masked.			
Interrupt RAW STATUS register						
<i>Address : CF00_021Ch</i>						
[31:9]	R	RESERVED	Reserved	0x00000000		
[8]		DECERR	DECERR from L3 HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[7]		SLVERR	SLVERR from L3 HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[6]		ERRRD	Error on L2 data RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[5]		ERRRT	Error on L2 tag RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[4]		ERRWD	Error on L2 data RAM, Write HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[3]		ERRWT	Error on L2 tag RAM, Write HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[2]		PARRD	Parity Error on L2 data RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[1]		PARRT	Parity Error on L2 tag RAM, Read HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
[0]		ECNTR	Even Counter1 and Event Counter 0 Overflow Increment HIGH: If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW: If the bits read LOW, no interrupt has been generated.			
Interrupt CLEAR register						
<i>Address : CF00_0220h</i>						
[31:9]	W	RESERVED	Reserved	0x00000000		
[8]		DECERR	DECERR from L3 When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
[7]		SLVERR	SLVERR from L3 When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
[6]		ERRRD	Error on L2 data RAM, Read When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
[5]		ERRRT	Error on L2 tag RAM, Read When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register.			

Bit	R/W	Symbol	Description	Reset Value		
			When a bit is written as 0, it has no effect.			
[4]		ERRWD	Error on L2 data RAM, Write When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
[3]		ERRWT	Error on L2 tag RAM, Write When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
[2]		PARRD	Parity Error on L2 data RAM, Read When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
[1]		PARRT	Parity Error on L2 tag RAM, Read When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
[0]		ECNTR	Even Counter1 and Event Counter 0 Overflow Increment When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect.			
CACHE SYNC						
Address : CF00_0730h						
[31:1]	R/W	RESERVED	Reserved	0x00000000		
[0]		CACHESYNC	CacheSYNC			
INVALIDATE line by pa						
Address : CF00_0770h						
[31:12]	R/W	TAG	Tag	0x00000000		
[11:5]		INDEX	Index			
[4:1]		RESERVED	Reserved			
[0]		CLEAN	Invalidate 0: Idle 1: Enable Invalidate			
invalidate by way						
Address : CF00_077Ch						
[31:28]	R/W	WAY	Way	0x00000000		
[27:12]		RESERVED	Reserved			
[11:5]		INDEX	Index			
[4:1]		RESERVED	Reserved			
[0]		CLEAN	Invalidate 0: Idle 1: Enable Invalidate			
clean line by pa						
Address : CF00_07B0h						
[31:12]	R/W	TAG	Tag	0x00000000		
[11:5]		INDEX	Index			
[4:1]		RESERVED	Reserved			
[0]		CLEAN	Invalidate			

Bit	R/W	Symbol	Description	Reset Value		
			0: Idle 1: Enable Invalidate			
clean line by set/way						
<i>Address : CF00_07B8h</i>						
[31:28]	R/W	WAY	Way	0x00000000		
[27:12]		RESERVED	Reserved			
[11:5]		INDEX	Index			
[4:1]		RESERVED	Reserved			
clean by way						
<i>Address : CF00_07BCCh</i>						
[31:16]	R/W	RESERVED	Reserved	0x00000000		
[15:0]		WAY BITS	Way Bits 8WAY: 0xFF 16WAY: 0xFFFF			
clean and invalidate line by pa						
<i>Address : CF00_07F0h</i>						
[31:12]	R/W	TAG	Tag	0x00000000		
[11:5]		INDEX	Index			
[4:1]		RESERVED	Reserved			
[0]		CLEAN	Invalidate 0: Idle 1: Enable Invalidate			
clean and 07F8h line by set/way						
<i>Address : CF00_07F0h</i>						
[31:28]	R/W	WAY	Way	0x00000000		
[27:12]		RESERVED	Reserved			
[11:5]		INDEX	Index			
[4:1]		RESERVED	Reserved			
clean and invalidate by way						
<i>Address : CF00_07FCCh</i>						
[31:16]	R/W	RESERVED	Reserved	0x00000000		
[15:0]		WAY BITS	Way Bits 8WAY: 0xFF 16WAY: 0xFFFF			
DATA Lockdown 0~7						
<i>Address : CF00_0900h, CF00_0908h</i>						
<i>Address : CF00_0910h, CF00_0918h</i>						
<i>Address : CF00_0920h, CF00_0928h</i>						
<i>Address : CF00_0930h, CF00_0938h</i>						
[31:16]	R/W	RESERVED	Reserved	0x00000000		
[15:0]		DATALOCK	Way Bits 8WAY: 0xFF			

Bit	R/W	Symbol	Description	Reset Value		
			16WAY: 0xFFFF			
InstructOn Lockdown 0~7						
<i>Address : CF00_0904h, CF00_090Ch</i>						
<i>Address : CF00_0914h, CF00_091Ch</i>						
<i>Address : CF00_0924h, CF00_092Ch</i>						
<i>Address : CF00_0934h, CF00_093Ch</i>						
[31:16]	R/W	RESERVED	Reserved	0x00000000		
[15:0]		INSTRLOCK	Way Bits 8WAY: 0xFF 16WAY: 0xFFFF			
lock down by line enable						
<i>Address : CF00_0950h</i>						
[31:1]	R/W	RESERVED	Reserved	0x00000000		
[0]		LOCKDOWN BY LINE ENB	0: Lockdown by line disabled. This is default 1: Lockdown by line enabled			
Unlock lines by way						
<i>Address : CF00_0950h</i>						
[31:16]	R/W	RESERVED	Reserved	0x00000000		
[15:0]		UNLOCK ALL LINES BY WAY	For all bits: 0: Unlock all lines disabled. This is the default 1: Unlock all lines operation in progress for the corresponding way.			
address filtering start						
<i>Address : CF00_0C00h</i>						
[31:20]	R/W	FILTERING START	Address filtering start address for bits[31:20] of the filtering address	0x00000000		
[19:1]		RESERVED	SBZ/RAZ			
[0]		FILTER ENB	Address filter enable			
address filtering end						
<i>Address : CF00_0C04h</i>						
[31:20]	R/W	FILTERING START	Address filtering start address for bits[31:20] of the filtering address	0x00000000		
[19:0]		RESERVED	SBZ/RAZ			
debug ctrl						
<i>Address : CF00_0F40h</i>						
[31:3]	R/W	RESERVED	Reserved	0x00000000		
[2]		SPNIDEN	Reads value of SPNIDEN Input.			
[1]		DWB	Disable Write-back force WT 0: Enable write-back behavior. This is the default. 1: Force write-through behavior			
[0]		DCL	Disable cache linefill 0: Enable cache line fills. This is the default 1: Disable cache linefills			
prefetch ctrl						
<i>Address : CF00_0F60h</i>						
[31]	R/W	RESERVED	Reserved	0x00000004		

Bit	R/W	Symbol	Description	Reset Value		
[30]		DOUBLE LINE FILL ENABLE	You can set the following options for this register bit: 0: The L2CC always issues 4x64-bit read bursts to L3 on reads that miss in the L2cache. This is the default. 1: The L2CC issues 8x64-bit read bursts to L3 on reads that miss in the L2 cache.			
[29]		INSTRUCTION PREFETCH ENABLE	You can set the following options for this register bit: 0: Instruction prefetching disabled. This is the default. 1: Instruction prefetching enabled.			
[28]		DATA PREFETCH ENABLE	You can set the following options for this register bit: 0 Data prefetching disabled. This is the default. 1: Data prefetching enabled.			
[27]		DOUBLE LINE FILL ON WRAP READ DISABLE	You can set the following options for this register bit: 0: Double linefill on WRAP read enabled. This is the default. 1: Double line fill on wrap read disabled.			
[26:25]		RESERVED	SBZ/RAZ			
[24]		PREFETCH DROP ENABLE	You can set the following options for this register bit: 0: The L2CC does not discard prefetch reads issued to L3. This is default 1: The L2CC discards prefetch reads issued to L3 when there is a resource conflict with explicit reads.			
[23]		INCR DOUBLE LINEFILL ENABLE	0: The L2CC does not issue INCR 8x64-bit read bursts to L3 on reads that miss in the L2 cache. This is the default. 1: The L2CC can issue INCR 8x64-bit read bursts to L3 on reads that miss in the L2cache.			
[22]		RESERVED	SBZ/RAZ			
[21]		NOT SAME ID ON EXCLUSIVE SEQUENCE ENABLE	You can set following options for this register bit: 0: Read and write portions of a non cacheable exclusive sequence have the same AXI ID when issued to L3. This is the default. 1: Read and write portions of a non cacheable exclusive sequence do not have the same AXI ID when issued to L3.			
[20:5]		RESERVED	SBZ/RAZ			
[4:0]		PREFETCH OFFSET	Default value = 0b00000 You must only use the prefetch offset values of 0-7, 15, 23, and 31 for these bits. The L2C-310 does not support the other values.			
power ctrl						
Address : CF00_0F80h						
[31:2]	R/W	RESERVED	Reserved	0x00000000		
[1]		DYNAMIC_CLK_GATING	Dynamic Clock gating Enable. 0: Disabled. This is the default. 1: Enabled			
[0]		STANBY MODE EN	Standby mode enable. 0: Disabled. This is the default. 1: Enabled.			

Section 7. *Secure JTAG*

7.1 Overview

The Secure JTAG consists of an Authentication & Authorization module an Access Provider. Secure JTAG Device support protection by user password can be unlocked by providing the correct password. Secure JTAG Connected with CoreSight at AHB AP. To activate the password unlock mechanism, the password exchange request must be applied to the AHB AP of CoreSight.

7.1.1 Features

- The Secure JTAG features: Authentication & Authorization Debug Module.

7.1.2 Block Diagram

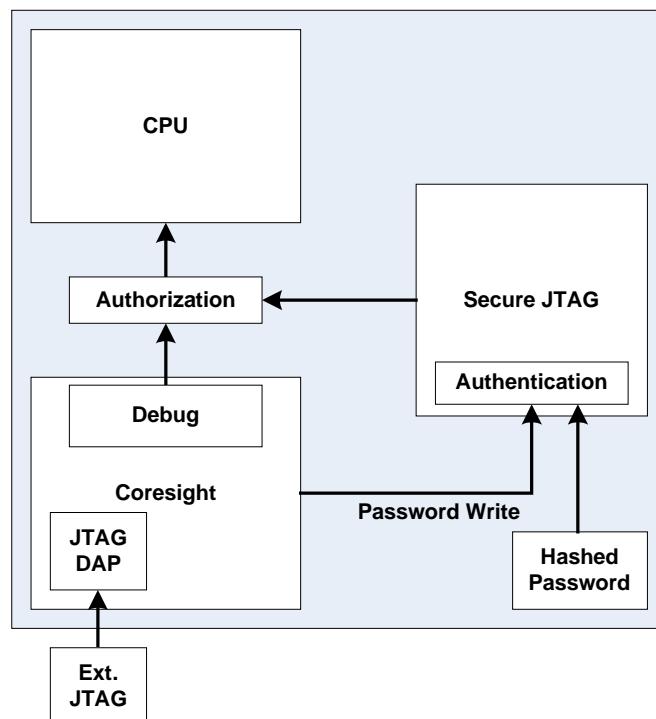


Figure 7-1. Secure JTAG Block Diagram

7.1.3 Secure JTAG User Config

- System L2 Cache Base Address: JTAG AHB-AP BASE Address 0x00000000

Section 8. DMA

8.1 Overview

The DMA Controller (DMAC) is an Advanced Microcontroller Bus Architecture (AMBA) block that connects to the Advanced High-performance Bus (AHB). There is an AHB slave interface for programming the DMAC and 2 AHB masters for data transfer. There are two DMAC in NXP4330D/Q and each DMAC has eight channels, which can buffer up to 4 words each. Each channel can transfer data through either of the AHB Master interfaces with the programmed data width and endianness. The DMAC supports 16 DMA requestors, and generates individually maskable interrupts for Terminal count and transfer error for each channel.

8.1.1 Features

- 16 DMA channels. Each channel can support a unidirectional transfer.
- 16 DMA requests. The DMAC provides 16 peripheral DMA request lines.
- Single DMA and burst DMA request signals. Each peripheral connected to the DMAC can assert either a burst DMA request or a single DMA request. You set the DMA burst size by programming the DMAC.
- Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral transfers.
- Scatter or gather DMA support through the use of linked lists.
- Hardware DMA channel priority. Each DMA channel has a specific hardware priority. DMA channel 0 has the highest priority and channel 7 has the lowest priority. If requests from two channels become active at the same time, the channel with the highest priority is serviced first.
- AHB slave DMA programming interface. You program the DMAC by writing to the DMA control registers over the AHB slave interface.
- Two AHB bus masters for transferring data. Use these interfaces to transfer data when a DMA request goes active.
- Incrementing or non-incrementing addressing for source and destination.
- Programmable DMA burst size. You can programme the DMA burst size to transfer data more efficiently. The burst size is usually set to half the size of the FIFO in the peripheral.
- Internal four word FIFO per channel.
- Supports eight, 16, and 32-bit wide transactions.
- Big-endian and little-endian support. The DMAC defaults to little-endian mode on reset.
- Separate and combined DMA error and DMA count interrupt requests. You can generate an interrupt to the processor on a DMA error or when a DMA count has reached 0. This is usually used to indicate that a transfer has finished. There are three interrupt request signals to do this:
 - **DMACINTTC** signals when a transfer has completed.
 - **DMACINTERR** signals when an error has occurred.
 - **DMACINTR** combines both the **DMACINTTC** and **DMACINTERR** interrupt request signals. You can use the **DMACINTR** interrupt request in systems that have few interrupt controller request inputs.
- Interrupt masking. You can mask the DMA error and DMA terminal count interrupt requests.
- Raw interrupt status. You can read the DMA error and DMA count raw interrupt status prior to masking.
- Test registers for use in block and integration system level testing.

- Identification registers that uniquely identify the DMAC. An operating system can use these to automatically configure itself.

8.1.2 Block Diagram

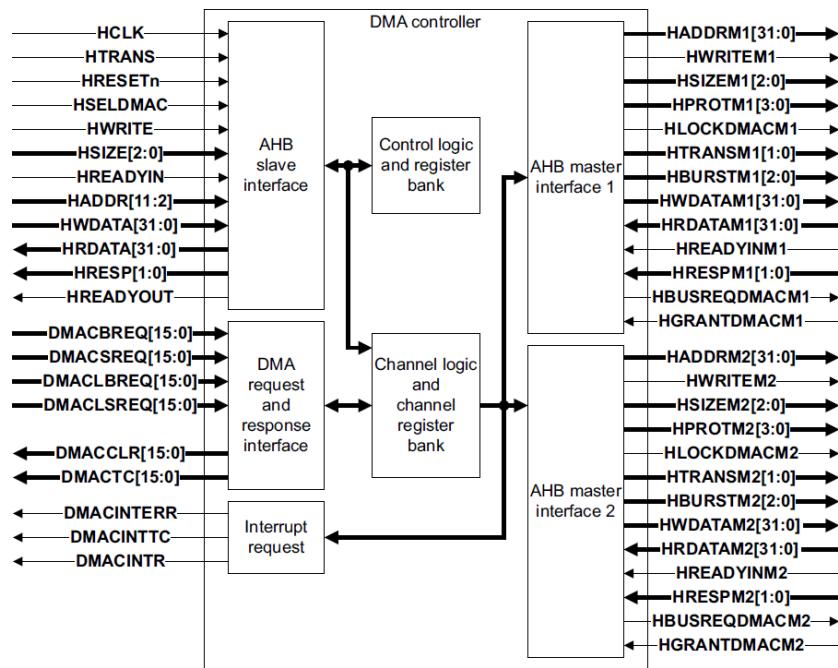


Figure 8-1. DMAC block diagram

8.2 Functional Description

8.2.1 Software considerations

You must take into account the following software considerations when programming the DMAC:

- There must not be any write-operation to Channel registers in an active channel after the channel enable is made HIGH. If you must reprogram any DMAC channel parameters, you must reprogram after disabling the DMAC channel.
- If the source width is less than the destination width, the TransferSize value multiplied by the source width must be an integral multiple of the destination width.
- When the source peripheral is the flow controller and the source width is less than the destination width, the number of transfers that the source peripheral performs, before asserting an **DMACLSREQ** or **DMACLBREQ**, must be so that the number of transfers multiplied by the source width is an integral multiple of the destination width. If this case is violated, the data can get stuck and lost in the FIFO causing UNPREDICTABLE results. You can abort the transfer by disabling the relevant DMAC channel.
- You must not program the SrcPeripheral and DestPeripheral bit fields in the DMACCxConfig Register with any value greater than 15. See *Channel Configuration Registers* on page 3-27.
- The SWidth and DWidth bit fields in the DMACCxControl Register must not indicate more than a 32-bit wide peripheral. See *Channel Control Registers* on page 3-23.
- After the software disables a channel by clearing the ChannelEnable bit in the DMACCxConfig Register, see *Channel Configuration Registers* on page 3-27, it must re-enable the bit only after it has polled a 0 in the corresponding DMACEnbldChns Register bit, see *Enabled Channel Register* on page 3-14. This is because the actual disabling does not immediately happen with the clearing of ChannelEnable bit. You must accommodate the latency of the ongoing AHB burst.
- The LLI field in the DMACCxLLIReg Register must not indicate an address greater than 0xFFFFFFFF0, otherwise the four-word LLI burst wraps over at 0x00000000 and the LLI data structure is not in contiguous memory locations. See *Channel Linked List Item Registers* on page 3-22.
- When the transfer size programmed in the DMAC is greater than the depth of the FIFO in a source or destination peripheral, you must only program the DMAC for non-incrementing address generation.
- A peripheral is expected to deassert any **DMACSREQ**, **DMACBREQ**, **DMACLSREQ**, or **DMACLBREQ** signals on receiving the **DMACCLR** signal irrespective of the request the **DMACCLR** was asserted in response to. This is because **DMACCLR** is not specific to a single-request signal, **DMACSREQ**, or burst-request signal, **DMACSBEQ**. The handshaking of **DMACCLR** is achieved with a logical OR of all the DMA requests in the DMAC.
- Note) It is illegal for a peripheral to give a new **DMACSREQ** or **DMACBREQ** signal while **DMACCLR** is HIGH.
- If you program the TransferSize field in the DMACCxControl Register, see *Channel Control Registers* on page 3-23, as zero, and the DMAC is the flow controller, the TransferSize field has no meaning in other flow-control modes, then the channel does not initiate any transfers. It is your responsibility to disable the channel by writing into the channel enable bit of the DMACCxConfig Register and reprogramming the channel again.

- You must not run the normal read-write tests on the DMACCxControl Register, see *Channel Control Registers* on page 3-23, because the TransferSize field is not a typical write and read-back register field. While writing, the TransferSize bit-field is like a control register because it determines how many transfers the DMAC performs. However, during read-back, TransferSize behaves like a status register because it returns the number of remaining transfers in terms of source width. So when TransferSize is read back, it returns the number of destination-transfer-completed stored in a separate counter called TrfSizeDst multiplied by a factor. The same physical register is not being written into and read from, and normal write and read-back tests are not applicable.
- In the destination flow control mode, with peripheral-to-peripheral transfer, if sufficient data is present in the channel FIFO to service a **DMACLSREQ** or **DMACLBREQ** request raised by a destination peripheral without requiring data to be fetched from the source peripheral, then the source peripheral is issued a **DMACTC**.
- For destination flow controlled case, peripheral-to-peripheral transfer, with DWidth < SWidth, the number of data bytes requested by the destination peripheral must be an integral multiple of Swidth expressed in bytes. If you do not ensure this, then the DMAC might fetch more data from the source peripheral than is required. This can result in data loss.
- At the end of accesses corresponding to low-priority channels, an IDLE cycle is inserted on the AHB bus to enable other masters to access the bus. This ensures that a low-priority channel does not monopolize the bus. It does, however, mean that the bus might be occupied by transactions corresponding to a low priority for up to 16 cycles in the worst case. This applies to all transfer configurations, including memory-to-memory transfers.

8.2.2 Programmer's Model

8.2.2.1 About the programmer's model

The DMAC enables the following types of transactions:

- memory-to-memory
- memory-to-peripheral
- peripheral-to-memory
- peripheral-to-peripheral.

Each DMA stream is configured to provide unidirectional DMA transfers for a single source and destination.

For example, a bidirectional serial port requires one stream for transmit and one for receive. The source and destination areas can each be either a memory region or a peripheral, and you can access them through the same AHB master, or one area by each master.

The base address of the DMAC is not fixed, and can be different for any particular system implementation. However, the offset of any particular register from the base address is fixed.

Register fields

The following applies to the registers that the DMAC uses:

- You must not access reserved or unused address locations because this can result in unpredictable behavior of the device.

- You must write reserved or unused bits of registers as zero, and ignore them on read unless otherwise stated in the relevant text.
- A system or power-on reset resets all register bits to a logic 0 unless otherwise stated in the relevant text.
- All registers support read and write accesses unless otherwise stated in the relevant text. A write updates the contents of a register, and a read returns the contents of the register.
- You can only access registers defined in this document using word reads and word writes, unless otherwise stated in the relevant text.

8.2.2.2 Programming the DMAC

Enabling the DMAC

Enable the DMAC by setting the DMA Enable, E, bit in the DMACConfiguration Register. See *Configuration Register* on page.

Disabling DMAC

To disable the DMAC:

- 1) Read the DMACEnblChns Register and ensure that you have disabled all the DMA channels. If any channels are active, see *Disabling a DMA channel* on page.
- 2) Disable the DMAC by writing 0 to the DMA Enable bit in the DMACConfiguration Register. See *Configuration Register* on page.

Enabling a DMA channel

Enable the DMA channel by setting the Channel Enable bit in the relevant DMA channel Configuration Register. See *Channel Configuration Registers* on page 3-27.

Note You must fully initialize the channel before you enable it. Additionally, you must set the Enable bit of the DMAC before you enable any channels.

Disabling a DMA channel

You can disable a DMA channel in the following ways:

- 1) Write directly to the Channel Enable bit.

Note You lose any outstanding data in the FIFOs if you use this method.

- 2) Use the Active and Halt bits in conjunction with the Channel Enable bit.

- 3) Wait until the transfer completes. The channel is then automatically disabled.

Disabling a DMA channel and losing data in the FIFO

Clear the relevant Channel Enable bit in the relevant channel Configuration Register.

See *Channel Configuration Registers* on page . The current AHB transfer, if one is in progress, completes and the channel is disabled.

Note You lose any data in the FIFO.

Disabling a DMA channel without losing data in the FIFO

To disable a DMA channel without losing data in the FIFO:

- 1) Set the Halt bit in the relevant channel Configuration Register. See *Channel Configuration Registers* on page 3-27. This causes any subsequent DMA requests to be ignored.
- 2) Poll the Active bit in the relevant channel Configuration Register until it reaches 0. This bit indicates whether there is any data in the channel that has to be transferred.
- 3) Clear the Channel Enable bit in the relevant channel Configuration Register.

Setting up a new DMA transfer

To set up a new DMA transfer:

- 1) If the channel is not set aside for the DMA transaction:
 - a. Read the DMACEnbldChns Register and determine the channels that are inactive. See *Enabled Channel Register* on page 3-14.
 - b. Choose an inactive channel that has the necessary priority.
- 2) Program the DMAC.

Halting a DMA channel

Set the Halt bit in the relevant DMA channel Configuration Register. The current source request is serviced. Any subsequent source DMA requests are ignored until the Halt bit is cleared.

Programming a DMA channel

To program a DMA channel:

- 1) Choose a free DMA channel with the necessary priority. DMA channel 0 has the highest priority and DMA channel 7 has the lowest priority.
- 2) Clear any pending interrupts on the channel you want to use by writing to the DMACIntTCClear and DMACIntErrClr Registers. See *Interrupt Terminal Count Clear Register* on page 3-11 and *Interrupt Error Clear Register* on page 3-12. The previous channel operation might have left interrupts active.
- 3) Write the source address into the DMACCxSrcAddr Register. See *Channel Source Address Registers* on page 3-21.
- 4) Write the destination address into the DMACCxDestAddr Register. See *Channel Destination Address Registers* on page 3-21.
- 5) Write the address of the next LLI into the DMACCxLLI Register. See *Channel Linked List Item Registers* on page 3-22. If the transfer consists of a single packet of data, you must write 0 into this register.
- 6) Write the control information into the DMACCxControl Register. See *Channel Control Registers* on page 3-23.
- 7) Write the channel configuration information into the DMACCxConfiguration Register. See *Channel Configuration Registers* on page 3-27. If the Enable bit is set, then the DMA channel is automatically enabled.

8.2.2.3 Register Name Description

Name	Address (0xC0000000)	Type	Reset value	Description
DMACINTSTATUS	0x000	RO	0x00	Interrupt Status Register

DMACINTTCSTATUS	0x004	RO	0x00	Interrupt Terminal Count Status Register
DMACINTTCCLEAR	0x008	WO	-	Interrupt Terminal Count Clear Register
DMACINTERRORSTATUS	0x00C	RO	0x00	Interrupt Error Status Register
DMACINTERRCLR	0x010	WO	-	Interrupt Error Clear Register
DMACRAWINTTCSTATUS	0x014	RO	-	Raw Interrupt Terminal Count Status Register
DMACRAWINTERRORSTATUS	0x018	RO	-	Raw Error Interrupt Status Register
DMACENBLDCHNS	0x01C	RO	0x00	Enabled Channel Register
DMACSOFTBREQ	0x020	R/W	0x0000	Software Burst Request Register
DMACSOFTSREQ	0x024	R/W	0x0000	Software Single Request Register
DMACSOFTLBREQ	0x028	R/W	0x0000	Software Last Burst Request Register
DMACSOFTLSREQ	0x02C	R/W	0x0000	Software Last Single Request Register
DMACCONFIGURATION	0x030	R/W	0b000	Configuration Register
DMACSYNC	0x34	R/W	0x0000	Synchronization Register
DMACC0SRCADDR	0x100	R/W	0x00000000	Channel Source Address Registers
DMACC0DESTADDR	0x104	R/W	0x00000000	Channel Destination Address Registers
DMACC0LLI	0x108	R/W	0x00000000	Channel Linked List Item Registers
DMACC0CONTROL	0x10C	R/W	0x00000000	Channel Control Registers
DMACC0CONFIGURATION	0x110	R/W	0x0000	Channel Configuration Registers
DMACC1SRCADDR	0x120	R/W	0x00000000	Channel Source Address Registers
DMACC1DESTADDR	0x124	R/W	0x00000000	Channel Destination Address Registers
DMACC1LLI	0x128	R/W	0x00000000	Channel Linked List Item Registers
DMACC1CONTROL	0x12C	R/W	0x00000000	Channel Control Registers
DMACC1CONFIGURATION	0x130	R/W	0x0000	Channel Configuration Registers
DMACC2SRCADDR	0x140	R/W	0x00000000	Channel Source Address Registers
DMACC2DESTADDR	0x144	R/W	0x00000000	Channel Destination Address Registers
DMACC2LLI	0x148	R/W	0x00000000	Channel Linked List Item Registers
DMACC2CONTROL	0x14C	R/W	0x00000000	Channel Control Registers
DMACC2CONFIGURATION	0x150	R/W	0x0000	Channel Configuration Registers
DMACC3SRCADDR	0x160	R/W	0x00000000	Channel Source Address Registers
DMACC3DESTADDR	0x164	R/W	0x00000000	Channel Destination Address Registers
DMACC3LLI	0x168	R/W	0x00000000	Channel Linked List Item Registers
DMACC3CONTROL	0x16C	R/W	0x00000000	Channel Control Registers
DMACC3CONFIGURATION	0x170	R/W	0x0000	Channel Configuration Registers
DMACC4SRCADDR	0x180	R/W	0x00000000	Channel Source Address Registers
DMACC4DESTADDR	0x184	R/W	0x00000000	Channel Destination Address Registers
DMACC4LLI	0x188	R/W	0x00000000	Channel Linked List Item Registers
DMACC4CONTROL	0x18C	R/W	0x00000000	Channel Control Registers
DMACC4CONFIGURATION	0x190	R/W	0x0000	Channel Configuration Registers
DMACC5SRCADDR	0x1A0	R/W	0x00000000	Channel Source Address Registers
DMACC5DESTADDR	0x1A4	R/W	0x00000000	Channel Destination Address Registers

DMACC5LLI	0x1A8	R/W	0x00000000	Channel Linked List Item Registers
DMACC5CONTROL	0x1AC	R/W	0x00000000	Channel Control Registers
DMACC5CONFIGURATION	0x1B0	R/W	0x0000	Channel Configuration Registers
DMACC6SRCADDR	0x1C0	R/W	0x00000000	Channel Source Address Registers
DMACC6DESTADDR	0x1C4	R/W	0x00000000	Channel Destination Address Registers
DMACC6LLI	0x1C8	R/W	0x00000000	Channel Linked List Item Registers
DMACC6CONTROL	0x1CC	R/W	0x00000000	Channel Control Registers
DMACC6CONFIGURATION	0x1D0	R/W	0x0000	Channel Configuration Registers
DMACC7SRCADDR	0x1E0	R/W	0x00000000	Channel Source Address Registers
DMACC7DESTADDR	0x1E4	R/W	0x00000000	Channel Destination Address Registers
DMACC7LLI	0x1E8	R/W	0x00000000	Channel Linked List Item Registers
DMACC7CONTROL	0x1EC	R/W	0x00000000	Channel Control Registers
DMACC7CONFIGURATION	0x1F0	R/W	0x0000	Channel Configuration Registers

Table 8-1. DMAC0 register summary

Name	Address (0xC0001000)	Type	Reset value	Description
DMACINTSTATUS	0x000	RO	0x00	Interrupt Status Register
DMACINTTCSTATUS	0x004	RO	0x00	Interrupt Terminal Count Status Register
DMACINTTCCLEAR	0x008	WO	-	Interrupt Terminal Count Clear Register
DMACINTERRORSTATUS	0x00C	RO	0x00	Interrupt Error Status Register
DMACINTERRCLR	0x010	WO	-	Interrupt Error Clear Register
DMACRAWINTTCSTATUS	0x014	RO	-	Raw Interrupt Terminal Count Status Register
DMACRAWINTERRORSTATUS	0x018	RO	-	Raw Error Interrupt Status Register
DMACENBLDCHNS	0x01C	RO	0x00	Enabled Channel Register
DMACSOFTBREQ	0x020	R/W	0x0000	Software Burst Request Register
DMACSOFTSREQ	0x024	R/W	0x0000	Software Single Request Register
DMACSOFTLBREQ	0x028	R/W	0x0000	Software Last Burst Request Register
DMACSOFTLSREQ	0x02C	R/W	0x0000	Software Last Single Request Register
DMACCONFIGURATION	0x030	R/W	0b000	Configuration Register
DMACSYNC	0x34	R/W	0x0000	Synchronization Register
DMACC0SRCADDR	0x100	R/W	0x00000000	Channel Source Address Registers
DMACC0DESTADDR	0x104	R/W	0x00000000	Channel Destination Address Registers
DMACC0LLI	0x108	R/W	0x00000000	Channel Linked List Item Registers
DMACC0CONTROL	0x10C	R/W	0x00000000	Channel Control Registers
DMACC0CONFIGURATION	0x110	R/W	0x0000	Channel Configuration Registers
DMACC1SRCADDR	0x120	R/W	0x00000000	Channel Source Address Registers
DMACC1DESTADDR	0x124	R/W	0x00000000	Channel Destination Address Registers
DMACC1LLI	0x128	R/W	0x00000000	Channel Linked List Item Registers
DMACC1CONTROL	0x12C	R/W	0x00000000	Channel Control Registers
DMACC1CONFIGURATION	0x130	R/W	0x0000	Channel Configuration Registers

DMACC2SRCADDR	0x140	R/W	0x00000000	Channel Source Address Registers
DMACC2DESTADDR	0x144	R/W	0x00000000	Channel Destination Address Registers
DMACC2LLI	0x148	R/W	0x00000000	Channel Linked List Item Registers
DMACC2CONTROL	0x14C	R/W	0x00000000	Channel Control Registers
DMACC2CONFIGURATION	0x150	R/W	0x0000	Channel Configuration Registers
DMACC3SRCADDR	0x160	R/W	0x00000000	Channel Source Address Registers
DMACC3DESTADDR	0x164	R/W	0x00000000	Channel Destination Address Registers
DMACC3LLI	0x168	R/W	0x00000000	Channel Linked List Item Registers
DMACC3CONTROL	0x16C	R/W	0x00000000	Channel Control Registers
DMACC3CONFIGURATION	0x170	R/W	0x0000	Channel Configuration Registers
DMACC4SRCADDR	0x180	R/W	0x00000000	Channel Source Address Registers
DMACC4DESTADDR	0x184	R/W	0x00000000	Channel Destination Address Registers
DMACC4LLI	0x188	R/W	0x00000000	Channel Linked List Item Registers
DMACC4CONTROL	0x18C	R/W	0x00000000	Channel Control Registers
DMACC4CONFIGURATION	0x190	R/W	0x0000	Channel Configuration Registers
DMACC5SRCADDR	0x1A0	R/W	0x00000000	Channel Source Address Registers
DMACC5DESTADDR	0x1A4	R/W	0x00000000	Channel Destination Address Registers
DMACC5LLI	0x1A8	R/W	0x00000000	Channel Linked List Item Registers
DMACC5CONTROL	0x1AC	R/W	0x00000000	Channel Control Registers
DMACC5CONFIGURATION	0x1B0	R/W	0x0000	Channel Configuration Registers
DMACC6SRCADDR	0x1C0	R/W	0x00000000	Channel Source Address Registers
DMACC6DESTADDR	0x1C4	R/W	0x00000000	Channel Destination Address Registers
DMACC6LLI	0x1C8	R/W	0x00000000	Channel Linked List Item Registers
DMACC6CONTROL	0x1CC	R/W	0x00000000	Channel Control Registers
DMACC6CONFIGURATION	0x1D0	R/W	0x0000	Channel Configuration Registers
DMACC7SRCADDR	0x1E0	R/W	0x00000000	Channel Source Address Registers
DMACC7DESTADDR	0x1E4	R/W	0x00000000	Channel Destination Address Registers
DMACC7LLI	0x1E8	R/W	0x00000000	Channel Linked List Item Registers
DMACC7CONTROL	0x1EC	R/W	0x00000000	Channel Control Registers
DMACC7CONFIGURATION	0x1F0	R/W	0x0000	Channel Configuration Registers

Table 8-2. DMAC1 register summary

8.2.2.4 Peripheral DMA Request ID

Index	Description	Index	Description
0	UART1 Tx	16	I2S2 Tx
1	UART1 Rx	17	I2S2 Rx
2	UART0 Tx	18	AC97 PCMOUT
3	UART0 Rx	19	AC97 PCMIN
4	UART2 Tx	20	AC97 MICIN
5	UART2 Rx	21	SPDIF RX

6	SSP0 Tx	22	SPDIF TX
7	SSP0 Rx	23	MPEGTSI0
8	SSP1 Tx	24	MPEGTSI1
9	SSP1 Rx	25	MPEGTSI2
10	SSP2 Tx	26	MPEGTSI4
11	SSP2 Rx	27	CRYPTO BR
12	I2S0 Tx	28	CRYPTO BW
13	I2S0 Rx	29	CRYPTO HR
14	I2S1 Tx	30	Reserved
15	I2S1 Rx	31	Reserved

Table 8-3. Peripheral DMA Request ID

8.2.2.5 Address generation

Address generation can be either incrementing or non-incrementing.

Note Address wrapping is not supported.

Bursts do not cross the 1KB address boundary.

8.2.2.6 Scatter / gather

Scatter/gather is supported through the use of linked lists. This means that the source and destination areas do not have to occupy contiguous areas in memory. You must set the DMACCxLLI Register to 0 if you do not require scatter/gather. For more information about scatter/gather DMA, see Appendix B *DMA Interface*.**Linked list items**

An LLI consists of four words. These words are organized in the following order:

- 1) DMACCxSrcAddr
- 2) DMACCxDestAddr
- 3) DMACCxLLI
- 4) DMACCxControl

Note The DMACCxConfiguration Channel Configuration Register is not part of the LLI.

Programming the DMAC for scatter/gather DMA

To program the DMAC for scatter/gather DMA:

- 1) Write the LLIs for the complete DMA transfer to memory. Each LLI contains four words:
 - source address
 - destination address
 - pointer to next LLI
 - control word.
 The last LLI has its linked list word pointer set to 0.
- 2) Choose a free DMA channel with the required priority. DMA channel 0 has the highest priority and DMA channel 7 the lowest priority.
- 3) Write the first LLI, previously written to memory, to the relevant channel in the DMAC.

4) Write the channel configuration information to the channel configuration register and set the Channel Enable bit. The DMAC then transfers the first and then subsequent packets of data as each LLI is loaded.

5) An interrupt can be generated at the end of each LLI depending on the Terminal Count bit in the DMACCxControl Register. If this bit is set, an interrupt is generated at the end of the relevant LLI. You must then service the interrupt request, and you must set the relevant bit in the DMAIntTCClear Register to clear the interrupt. If so, you must service this interrupt request and you must set the relevant IntTCClear bit in the DMAIntTCClr Register to clear the interrupt request interrupt.

Scatter/gather through linked list operation

A series of linked lists define the source and destination data areas. Each LLI controls the transfer of one block of data, and then optionally loads another LLI to continue the DMA operation, or stops the DMA stream. The first LLI is programmed into the DMAC.

The data to be transferred described by an LLI, referred to as the packet of data, usually requires one or more DMA bursts, to each of the source and destination.

Figure 8-2 shows an example of an LLI. A rectangle of memory must be transferred to a peripheral. The addresses of each line of data are given, in hexadecimal, at the left-hand side of the figure. The LLIs describing the transfer are to be stored contiguously from address 0x20000.

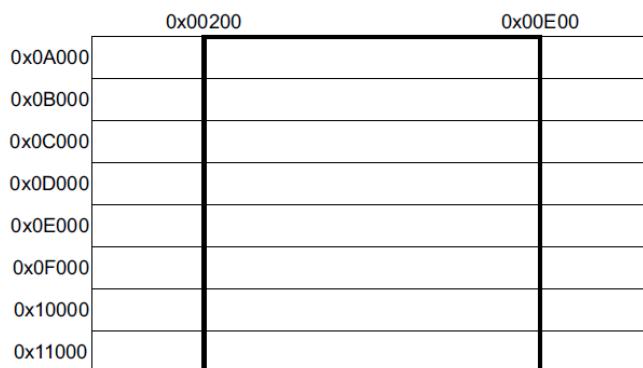


Figure 8-2 LLI example

The first LLI, stored at 0x20000, defines the first block of data to be transferred. This is the data stored between addresses 0xA200 and 0xAE00:

- source start address 0xA200
- destination address set to the destination peripheral address
- transfer width, word, 32-bit
- transfer size, 3072 bytes, 0xC00
- source and destination burst sizes, 16 transfers
- next LLI address, 0x20010.

The second LLI, stored at 0x20010, defines the next block of data to be transferred:

- source start address 0xB200
- destination address set to the destination peripheral address
- transfer width, word, 32-bit
- transfer size, 3072 bytes, 0xC00
- source and destination burst sizes, 16 transfers
- next LLI address, 0x20020.

A chain of descriptors is built up, each one pointing to the next in the series. To initialize the DMA stream, the first LLI, 0x20000, is programmed into the DMAC. When the first packet of data has been transferred, the next LLI is automatically loaded.

The final LLI is stored at 0x20070 and contains:

- source start address 0x11200
- destination address set to the destination peripheral address
- transfer width, word, 32-bit
- transfer size, 3072 bytes, 0xC00
- source and destination burst sizes, 16 transfers
- next LLI address, 0x0.

Because the next LLI address is set to zero, this is the last descriptor, and the DMA channel is disabled after transferring the last item of data. The channel is probably set to generate an interrupt at this point to indicate to the ARM processor that the channel can be reprogrammed.

8.2.2.7 Interrupt requests

Interrupt requests can be generated when an AHB error is encountered, or at the end of a transfer, terminal count, after all the data corresponding to the current LLI has been transferred to the destination. The interrupts can be masked by programming the relevant bits on the relevant DMACCxControl and DMACCxConfiguration Channel Registers.

Interrupt Status Registers are provided. They group the interrupt requests from all the DMA channels prior to interrupt masking, DMACRawIntTCStatus, DMACRawIntErrorStatus, and after interrupt masking, DMACIntTCStatus, DMACIntErrorStatus.

The DMACIntStatus Register combines both the DMACIntTCStatus and DMACIntErrorStatus requests into a single register to enable the source of an interrupt to be found quickly. Writing to the DMACIntTCClear or the DMACIntErrClr Registers with a bit set HIGH enables selective clearing of interrupts.

The DMAC provides two interrupt request connection schemes. See *Interrupt controller connectivity* on page 2-16. The simplest connection scheme has a combined error and end of transfer complete interrupt request. To find the source of an interrupt, you must read both the DMACIntStatus and DMACIntTCStatus Registers.

For faster interrupt response, you can use an alternate connection scheme. This scheme uses separate interrupt requests for the error and transfer complete requests. Read either the DMACIntTCStatus or DMACIntErrorStatus Registers to find the source of an interrupt.

Combined terminal count and error interrupt sequence flow

When you use the **DMACINTR** interrupt request:

- 1) You must wait until the combined interrupt request from the DMAC goes active. Assuming the interrupt is enabled in the interrupt controller and in the processor, the processor branches to the interrupt vector address and enters the interrupt service routine.
- 2) You must read the interrupt controller Status Register and determine whether the source of the request was the DMAC.
- 3) You must read the DMACIntStatus Register to determine the channel that generated the interrupt. If more than one request is active, it is recommended that you check the highest priority channels first.
- 4) You must read the DMACIntTCStatus Register to determine whether the interrupt was generated because of the end of the transfer, terminal count, or because an error occurred. A HIGH bit indicates that the transfer completed.
- 5) You must read the DMACIntErrorStatus Register to determine whether the interrupt was generated because of the

end of the transfer, terminal count, or because an error occurred. A HIGH bit indicates that an error occurred.

- 6) You must write a 1 to the relevant bit in the DMACIntTCClear, or DMACIntErrClr, Register to clear the interrupt request.

Terminal count interrupt sequence flow

When the separate, **DMACINTTC** and **DMACINTERR**, interrupt requests are used:

- 1) You must wait until the terminal count DMA interrupt request goes active. Assuming the interrupt is enabled in the interrupt controller and in the processor, the processor branches to the interrupt vector address and enters the interrupt service routine.
- 2) You must read the interrupt controller Status Register to determine if the source of the interrupt request was the DMAC asserting the **DMACINTTC** signal.
- 3) You must read the DMACIntTCStatus Register to determine the channel that generated the interrupt. If more than one request is active, it is recommended that you service the highest priority channel first.
- 4) You must service the interrupt request.
- 5) You must write a 1 to the relevant bit in the DMACIntTCClear Register to clear the interrupt request.

Error interrupt sequence flow

When the separate interrupt requests, **DMACINTTC** and **DMACINTERR**, are used:

- 1) You must wait until the interrupt request goes active because of a DMA channel error. Assuming the interrupt is enabled in the interrupt controller and in the processor, the processor branches to the interrupt vector address and enters the interrupt service routine.
- 2) You must read the Interrupt Controllers Status Register to determine if the source of the request was the DMAC asserting the **DMACINTERR** signal.
- 3) You must read the DMACIntErrorStatus Register to determine the channel that generated the interrupt. If more than one request is active it is recommended that you check the highest priority channels first.
- 4) You must service the interrupt request.
- 5) You must write a 1 to the relevant bit in the DMACIntErrClr Register to clear the interrupt request.

Interrupt polling sequence flow

The DMAC interrupt request signal is masked out, disabled in the interrupt controller, or disabled in the processor. When polling the DMAC, you must:

- 1) Read the DMACIntStatus Register. If none of the bits are HIGH repeat this step, otherwise, go to step 2. If more than one request is active, it is recommended that you check the highest priority channels first.
- 2) Read the DMACIntTCStatus Register to determine if the interrupt was generated because of the end of the transfer, terminal count, or because of error occurred. A HIGH bit indicates that the transfer completed.
- 3) Service the interrupt request.
- 4) For an error interrupt, write a 1 to the relevant bit of the DMACIntErrClr Register to clear the interrupt request. For a terminal count interrupt, write a 1 to the relevant bit of the DMACIntTCClr Register.

8.2.2.8 DMAC data flow

Memory-to-memory DMA flow

For a memory-to-memory DMA flow:

- 1) Program and enable the DMA channel.
- 2) Transfer data whenever the DMA channel has the highest pending priority and the DMAC gains bus master ship of the AHB bus.
- 3) If an error occurs while transferring the data, generate an error interrupt and disable the DMA stream.
- 4) Decrement the transfer count.
- 5) If the count has reached zero:
 - a. Generate a terminal count interrupt. You can mask the interrupt.
 - b. If the DMACCxLLI Register is not 0, then reload the following registers and go to back to step 2:
 - DMACCxSrcAddr
 - DMACCxDestAddr
 - DMACCxLLI
 - DMACCxControl.
 - c. However, if DMACCxLLI is 0, the DMA stream is disabled and the flow sequence ends.

Memory-to-peripheral, or peripheral-to-memory DMA flow

For a peripheral-to-memory or memory-to-peripheral DMA flow:

- 1) Program and enable the DMA channel.
- 2) Wait for a DMA request.
- 3) The DMAC then starts transferring data when:
 - a. The DMA request goes active.
 - b. The DMA stream has the highest pending priority.
 - c. The DMAC is the bus master of the AHB bus.
- 4) If an error occurs while transferring the data, an error interrupt is generated and the DMA stream is disabled, and the flow sequence ends.
- 5) Decrement the transfer count if the DMAC is controlling the flow control.
- 6) If the transfer has completed, indicated by the transfer count reaching 0 if the DMAC is performing flow control, or by the peripheral setting the **DMACLBREQ** or **DMACLSREQ** signals if the peripheral is performing flow control:
 - a. The DMAC asserts the **DMACTC** signal.
 - b. The terminal count interrupt is generated. You can mask this interrupt.
 - c. If the DMACCxLLI Register is not 0, then reload the following registers and go to back to step 2:
 - DMACCxSrcAddr
 - DMACCxDestAddr
 - DMACCxLLI
 - DMACCxControl.

- However, if DMACCxLLI is 0, the DMA stream is disabled and the flow sequence ends.

Peripheral-to-peripheral DMA flow

For a peripheral-to-peripheral DMA flow:

- 1) Program and enable the DMA channel.
- 2) Wait for a source DMA request.
- 3) The DMAC then starts transferring data when:
 - a. The DMA request goes active.
 - b. The DMA stream has the highest pending priority.
 - c. The DMAC is the bus master of the AHB bus.
- 4) If an error occurs while transferring the data, an error interrupt is generated, then finishes.
- 5) Decrement the transfer count if the DMAC is controlling the flow control.
- 6) If the transfer has completed, indicated by the transfer count reaching 0 if the DMAC is performing flow control, or by the peripheral setting the **DMACLBREQ** or **DMACLSREQ** signals if the peripheral is performing flow control:
 - a. The DMAC asserts the **DMACTC** signal to the source peripheral.
 - b. Subsequent source DMA requests are ignored.
- 7) When the destination DMA request goes active and there is data in the DMAC FIFO, transfer data into the destination peripheral.
- 8) If an error occurs while transferring the data, an error interrupt is generated and the DMA stream is disabled, and the flow sequence ends.
- 9) If the transfer has completed, it is indicated by the transfer count reaching 0 if the DMAC is performing flow control, or by the peripheral setting the **DMACLBREQ** or **DMACLSREQ** signals if the peripheral is performing flow control. The following happens:
 - a. The DMAC asserts the **DMACTC** signal to the destination peripheral.
 - b. The terminal count interrupt is generated. You can mask this interrupt.
 - c. If the DMACCxLLI Register is not 0, then reload the following registers and go to back to step 2:
 - DMACCxSrcAddr
 - DMACCxDestAddr
 - DMACCxLLI
 - DMACCxControl.
 - However, if DMACCxLLI is 0, the DMA stream is disabled and the flow sequence ends.

8.3 Register Summary

This section describes the DMAC registers.

Bit	R/W	Symbol	Description
Interrupt Status Register			
The read-only DMACIntStatus Register, with address offset of 0x000, shows the status of the interrupts after masking. A HIGH bit indicates that a specific DMA channel interrupt request is active. You can generate the request from either the error or terminal count interrupt requests.			
[31:8]	R	SESREQSCS	Read undefined
[7:0]	R	INTSTATUS	Status of the DMA interrupts after masking
Interrupt Terminal Count Status Register			
The read-only DMACIntTCStatus Register, with address offset of 0x004, indicates the status of the terminal count after masking. You must use this register in conjunction with the DMACIntStatus Register if you use the combined interrupt request, DMACINTR, to request interrupts. If you use the DMACINTTC interrupt request, then you only have to read the DMACIntTCStatus Register to ascertain the source of the interrupt request.			
[31:8]	R	-	Read undefined
[7:0]	R	INTTCSTATUS	Interrupt terminal count request status
Interrupt Terminal Count Clear Register			
The write-only DMACIntTCClear Register, with address offset of 0x008, clears a terminal count interrupt request. When writing to this register, each data bit that is set HIGH causes the corresponding bit in the Status Register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.			
[31:8]	W	-	Undefined. Write as zero.
[7:0]	W	INTTCCLEAR	Terminal count request clear.
Interrupt Error Status Register			
The read-only DMACIntErrorStatus Register, with address offset of 0x00C, indicates the status of the error request after masking. You must use this register in conjunction with the DMACIntStatus Register if you use the combined interrupt request, DMACINTR, to request interrupts. If you use the DMACINTERR interrupt request, then only read the DMACIntErrorStatus Register.			
[31:8]	R	-	Read undefined
[7:0]	R	INTERRORSTATUS	Interrupt error status
Interrupt Error Clear Register			
The write-only DMACIntErrClr Register, with address offset of 0x010, clears the error interrupt requests. When writing to this register, each data bit that is HIGH causes the corresponding bit in the Status Register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.			
[31:8]	W	-	Undefined. Write as zero.
[7:0]	W	INTERRCLR	IntErrClr Interrupt error clear.
Raw Interrupt Terminal Count Status Register			
The read-only DMACRawIntTCStatus Register, with address offset of 0x014, indicates the DMA channels that are requesting a transfer complete, terminal count interrupt, prior to masking. A HIGH bit indicates that the terminal count interrupt request is active prior to masking.			
[31:8]	R	-	Read undefined
[7:0]	R	RAWINTTCSTATUS	Status of the terminal count interrupt prior to masking
Raw Error Interrupt Status Register			
The read-only DMACRawIntErrorStatus Register, with address offset of 0x018, indicates the DMA channels that are requesting an error interrupt prior to masking. A HIGH bit indicates that the error interrupt request is active prior to masking.			
[31:8]	R	-	Read undefined
[7:0]	R	RAWINTERRORSTATUS	Status of the error interrupt prior to masking
Enabled Channel Register			
The read-only DMACEnblChns Register, with address offset of 0x01C, indicates the DMA channels that are enabled, as indicated by the Enable bit in the DMAACCxConfiguration Register. A HIGH bit indicates that a DMA channel is enabled. A bit is cleared on completion of the DMA transfer.			
[31:8]	R	-	Read undefined
[7:0]	R	ENABLEDCHANNELS	Channel enable status

Bit	R/W	Symbol	Description
Software Burst Request Register			
The read/write DMACSoftBReq Register, with address offset of 0x020, enables DMA burst requests to be generated by software. You can generate a DMA request for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting DMA burst transfers. You can generate a request from either a peripheral or the software request register.			
[31:16]	R/W	-	Read undefined. Write as zero.
[15:0]	R/W	SOFTBREQ	Software burst request.
Software Single Request Register			
The read/write DMACSoftSReq Register, with address offset of 0x024, enables DMA single requests to be generated by software. You can generate a DMA request for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting single DMA transfers. You can generate a request from either a peripheral or the software request register.			
[31:16]	R/W	-	Read undefined. Write as zero.
[15:0]	R/W	SOFTSREQ	Software single request.
Software Last Burst Request Register			
The read/write DMACSoftLBReq Register, with address offset of 0x028, enables software to generate DMA last burst requests. You can generate a DMA request for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting last burst DMA transfers. You can generate a request from either a peripheral or the software request register.			
[31:16]	R/W	-	Read undefined. Write as zero.
[15:0]	R/W	SOFTLBREQ	Software last burst request.
Software Last Single Request Register			
The read/write DMACSoftLSReq Register, with address offset of 0x02C, enables software to generate DMA last single requests. You can generate a DMA request for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting last single DMA transfers. You can generate a request from either a peripheral or the software request register.			
[31:16]	R/W	-	Read undefined. Write as zero.
[15:0]	R/W	SOFTLSREQ	Software last single request.
Configuration Register			
The read/write DMACConfiguration Register, with address offset of 0x030, configures the operation of the DMAC. You can alter the endianness of the individual AHB master interfaces by writing to the M1 and M2 bits of this register. The M1 bit enables you to alter the endianness of AHB master interface 1. The M2 bit enables you to alter the endianness of AHB master interface 2. The AHB master interfaces are set to little-endian mode on reset.			
[31:3]	R/W	-	Read undefined. Write as zero.
[2]	R/W	M2	AHB Master 2 endianness configuration: 0 = little-endian mode 1 = big-endian mode. This bit is reset to 0.
[1]	R/W	M1	AHB Master 1 endianness configuration: 0 = little-endian mode 1 = big-endian mode. This bit is reset to 0.
[0]	R/W	E	DMAC enable: 0 = disabled 1 = enabled. This bit is reset to 0. Disabling the DMAC reduces power consumption.
Synchronization Register			
The read/write DMACSync Register, with address offset of 0x034, enables or disables synchronization logic for the DMA request signals. The DMA request signals consist of:			
<ul style="list-style-type: none"> • DMACBREQ[15:0] • DMACSREQ[15:0] • DMACLBREQ[15:0] • DMACLSREQ[15:0] 			
A bit set to 0 enables the synchronization logic for a particular group of DMA requests.			
A bit set to 1 disables the synchronization logic for a particular group of DMA requests.			
This register is reset to 0, and synchronization logic enabled.			

Bit	R/W	Symbol	Description
Note It is illegal for a peripheral to give a new DMACSREQ or DMACBREQ signal while DMACCLR is HIGH.			
Note You must use synchronization logic when the peripheral generating the DMA request runs on a different clock to the DMAC. For peripherals running on the same clock as the DMAC, disabling the synchronization logic improves the DMA request response time. If necessary, synchronize the DMA response signals, DMACCLR and DMACTC, in the peripheral.			
[31:16]	R/W	-	Read undefined. Write as zero.
[15:0]	R/W	DMACSYNC	DMA synchronization logic for DMA request signals enabled or disabled. A LOW bit indicates that the synchronization logic for the request signals is enabled. A HIGH bit indicates that the synchronization logic is disabled.
Channel registers The channel registers are for programming a DMA channel. These registers consist of: <ul style="list-style-type: none">• eight DMACCxSrcAddr Registers• eight DMACCxDestAddr Registers• eight DMACCxLLI Registers• eight DMACCxControl Registers• eight DMACCxConfiguration Registers. When performing scatter/gather DMA, the first four registers are automatically updated.			
Note Unpredictable behavior can result if you update the channel registers when a transfer is taking place. If you want to change the channel configurations, you must disable the channel first and then reconfigure the relevant register.			
Channel Source Address Registers The eight read/write DMACCxSrcAddr Registers, with address offsets of 0x100, 0x120, 0x140, 0x160, 0x180, 0x1A0, 0x1C0, and 0x1E0 respectively, contain the current source address, byte-aligned, of the data to be transferred. Software programs each register directly before the appropriate channel is enabled. When the DMA channel is enabled, this register is updated: <ul style="list-style-type: none">• as the source address is incremented• by following the linked list when a complete packet of data has been transferred. Reading the register when the channel is active does not provide useful information. This is because by the time the software has processed the value read, the channel might have progressed. It is intended to be read-only when the channel has stopped, and in such case, it shows the source address of the last item read.			
Note You must align source and destination addresses to the source and destination widths.			
[31:0]	R/W	SRCAADDR	DMA source address
Channel Destination Address Registers The eight read/write DMACCxDestAddr Registers, with address offsets of 0x104, 0x124, 0x144, 0x164, 0x184, 0x1A4, 0x1C4, and 0x1E4 respectively, contain the current destination address, byte-aligned, of the data to be transferred. Software programs each register directly before the channel is enabled. When the DMA channel is enabled, the register is updated as the destination address is incremented and by following the linked list when a complete packet of data has been transferred. Reading the register when the channel is active does not provide useful information. This is because by the time the software has processed the value read, the channel might have progressed. It is intended to be read-only when a channel has stopped. In this case, it shows the destination address of the last item read.			
[31:0]	R/W	DESTADDR	DMA destination address
Channel Linked List Item Registers The eight read/write DMACCxLLI Registers, with address offsets of 0x108, 0x128, 0x148, 0x168, 0x188, 0x1A8, 0x1C8, and 0x1E8 respectively, contain a word-aligned address of the next LLI. If the LLI is 0, then the current LLI is the last in the chain, and the DMA channel is disabled after all DMA transfers associated with it are completed.			
Note Programming this register when the DMA channel is enabled has unpredictable results.			
Note To make loading the LLIs more efficient for some systems, you can make the LLI data structures 4-word aligned.			
[31:2]		LLI	Linked list item. Bits [31:2] of the address for the next LLI. Address bits [1:0] are 0.
[1]	R/		Read undefined. Write as zero.
[0]		LM	AHB master select for loading the next LLI LM = 0 = AHB Master 1 LM = 1 = AHB Master 2.
Channel Control Registers The eight read/write DMACCxControl Registers, with address offsets of 0x010C, 0x12C, 0x14C, 0x16C, 0x18C, 0x1AC, 0x1CC, and 0x1EC respectively, contain DMA channel control information such as the transfer size, burst size, and transfer width. Software programs each register directly before the DMA channel is enabled. When the channel is enabled, the register is updated by following the linked list when a complete packet of data has been transferred. Reading the register while the channel is active does not give useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read-only when a channel has stopped.			

Bit	R/W	Symbol	Description												
Below Table lists the values of the DBSize or SBSize bits and their corresponding burst sizes.															
Below Table Source or destination burst size															
Bit value of DBSize or SBSize		Source or destination burst transfer request size													
0b000		1													
0b001		4													
0b010		8													
0b011		16													
0b100		32													
0b101		64													
0b110		128													
0b111		256													
Below Table lists the value of the SWidth or DWidth bits and their corresponding widths.															
Below Table lists the value of the SWidth or DWidth bits and their corresponding widths.															
Bit value of SWidth or DWidth		Source or destination width													
0b000		Byte, 8-bit													
0b001		Halfword, 16-bit													
0b010		Word, 32-bit													
0b011		Reserved													
0b100		Reserved													
0b101		Reserved													
0b110		Reserved													
0b111		Reserved													
Protection and access information															
AHB access information is provided to the source and destination peripherals when a transfer occurs. The transfer information is provided by programming the DMA channel, the Prot bit of the DMACCxControl Register, and the Lock bit of the DMACCxConfiguration Register. Software programs these bits, and peripherals can use this information if necessary. Three bits of information are provided. Below Table lists the purposes of the three protection bits.															
Below Table Protection bits															
<table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> <th>Purpose</th> </tr> </thead> <tbody> <tr> <td>[0]</td> <td>Privileged or User</td> <td>Indicates whether the access is in User, or Privileged mode: 0 = user mode 1 = privileged mode. This bit controls the AHB HPROT[1] signal.</td> </tr> <tr> <td>[1]</td> <td>Bufferable or Nonbufferable</td> <td>Indicates whether or not the access can be buffered: 0 = non-bufferable 1 = bufferable. This bit indicates whether or not the access is bufferable. For example, you can use this bit to indicate to an AMBA bridge that the read can complete in zero wait states on the source bus without waiting for it to arbitrate for the destination bus and for the slave to accept the data. This bit controls the AHB HPROT[2] signal.</td> </tr> <tr> <td>[2]</td> <td>Cacheable or Noncacheable</td> <td>Indicates whether or not the access can be cached: 0 = non-cacheable 1 = cacheable. This bit indicates whether or not the access is cacheable. For example, you can use this bit to indicate to an AMBA bridge that when it saw the first read of a burst of eight it can transfer the whole burst of eight reads on the destination bus, rather than pass the transactions through one at a time. This bit controls the AHB HPROT[3] signal.</td> </tr> </tbody> </table>				Bits	Description	Purpose	[0]	Privileged or User	Indicates whether the access is in User, or Privileged mode: 0 = user mode 1 = privileged mode. This bit controls the AHB HPROT[1] signal.	[1]	Bufferable or Nonbufferable	Indicates whether or not the access can be buffered: 0 = non-bufferable 1 = bufferable. This bit indicates whether or not the access is bufferable. For example, you can use this bit to indicate to an AMBA bridge that the read can complete in zero wait states on the source bus without waiting for it to arbitrate for the destination bus and for the slave to accept the data. This bit controls the AHB HPROT[2] signal.	[2]	Cacheable or Noncacheable	Indicates whether or not the access can be cached: 0 = non-cacheable 1 = cacheable. This bit indicates whether or not the access is cacheable. For example, you can use this bit to indicate to an AMBA bridge that when it saw the first read of a burst of eight it can transfer the whole burst of eight reads on the destination bus, rather than pass the transactions through one at a time. This bit controls the AHB HPROT[3] signal.
Bits	Description	Purpose													
[0]	Privileged or User	Indicates whether the access is in User, or Privileged mode: 0 = user mode 1 = privileged mode. This bit controls the AHB HPROT[1] signal.													
[1]	Bufferable or Nonbufferable	Indicates whether or not the access can be buffered: 0 = non-bufferable 1 = bufferable. This bit indicates whether or not the access is bufferable. For example, you can use this bit to indicate to an AMBA bridge that the read can complete in zero wait states on the source bus without waiting for it to arbitrate for the destination bus and for the slave to accept the data. This bit controls the AHB HPROT[2] signal.													
[2]	Cacheable or Noncacheable	Indicates whether or not the access can be cached: 0 = non-cacheable 1 = cacheable. This bit indicates whether or not the access is cacheable. For example, you can use this bit to indicate to an AMBA bridge that when it saw the first read of a burst of eight it can transfer the whole burst of eight reads on the destination bus, rather than pass the transactions through one at a time. This bit controls the AHB HPROT[3] signal.													
[31]	R	I	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.												
[30:28]		PROT	Protection.												
[27]	R	DI	Destination increment. When set, the destination address is incremented after each transfer.												
[26]	R	SI	Source increment. When set, the source address is incremented after each transfer.												
[25]		D	Destination AHB master select: 0 = AHB master 1 selected for the destination transfer 1 = AHB master 2 selected for the destination transfer.												
[24]		S	Source AHB master select: 0 = AHB master 1 selected for the source transfer 1 = AHB master 2 selected for the source transfer.												
[23:21]		DWIDTH	Destination transfer width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data when required.												

Bit	R/W	Symbol	Description																											
[20:18]		SWIDTH	Source transfer width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data when required.																											
[17:15]		DBSIZE	Destination burst size. Indicates the number of transfers that make up a destination burst transfer request. You must set this value to the burst size of the destination peripheral, or if the destination is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the DMACxBREQ signal goes active in the destination peripheral. The burst size is not related to the AHB HBURST signal.																											
[14:12]		SBSIZE	Source burst size. Indicates the number of transfers that make up a source burst. You must set this value to the burst size of the source peripheral, or if the source is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the DMACxBREQ signal goes active in the source peripheral. The burst size is not related to the AHB HBURST signal.																											
[11:0]	R/W	TRANSFERSIZE	<p>Transfer size. A write to this field sets the size of the transfer when the DMAC is the flow controller. This value counts down from the original value to zero, and so its value indicates the number of transfers left to complete. A read from this field provides the number of transfers still to be completed on the destination bus. Reading the register when the channel is active does not give useful information because by the time the software has processed the value read, the channel might have progressed. Only use it when a channel is enabled, and then disabled.</p> <p>Program the transfer size value to zero if the DMAC is not the flow controller. If you program the TransferSize to a non-zero value, the DMAC might attempt to use this value instead of ignoring the TransferSize.</p>																											
Channel Configuration Registers																														
The eight DMACCxConfiguration Registers, with address offsets of 0x110, 0x130, 0x150, 0x170, 0x190, 0x1B0, 0x1D0, and 0x1F0 respectively, are read/write and configure the DMA channel. The registers are not updated when a new LLI is requested.																														
Below Table lists the bit values of the three flow control and transfer type bits.																														
Below Table Flow control and transfer type bits																														
<table border="1"> <thead> <tr> <th>Bitvalue</th> <th>Transfer type</th> <th>Controller</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Memory-to-memory</td> <td>DMA</td> </tr> <tr> <td>001</td> <td>Memory-to-peripheral</td> <td>DMA</td> </tr> <tr> <td>010</td> <td>Peripheral-to-memory</td> <td>DMA</td> </tr> <tr> <td>011</td> <td>Source peripheral-to-destination peripheral</td> <td>DMA</td> </tr> <tr> <td>100</td> <td>Source peripheral-to-destination peripheral</td> <td>Destination peripheral</td> </tr> <tr> <td>101</td> <td>Memory-to-peripheral</td> <td>Peripheral</td> </tr> <tr> <td>110</td> <td>Peripheral-to-memory</td> <td>Peripheral</td> </tr> <tr> <td>111</td> <td>Source peripheral-to-destination peripheral</td> <td>Source peripheral</td> </tr> </tbody> </table>				Bitvalue	Transfer type	Controller	000	Memory-to-memory	DMA	001	Memory-to-peripheral	DMA	010	Peripheral-to-memory	DMA	011	Source peripheral-to-destination peripheral	DMA	100	Source peripheral-to-destination peripheral	Destination peripheral	101	Memory-to-peripheral	Peripheral	110	Peripheral-to-memory	Peripheral	111	Source peripheral-to-destination peripheral	Source peripheral
Bitvalue	Transfer type	Controller																												
000	Memory-to-memory	DMA																												
001	Memory-to-peripheral	DMA																												
010	Peripheral-to-memory	DMA																												
011	Source peripheral-to-destination peripheral	DMA																												
100	Source peripheral-to-destination peripheral	Destination peripheral																												
101	Memory-to-peripheral	Peripheral																												
110	Peripheral-to-memory	Peripheral																												
111	Source peripheral-to-destination peripheral	Source peripheral																												
[31:19]	-	-	Read undefined. Write as zero.																											
[18]	R/W	H	<p>Halt:</p> <p>0 = enable DMA requests</p> <p>1 = ignore extra source DMA requests.</p> <p>The contents of the channels FIFO are drained.</p> <p>You can use this value with the Active and Channel Enable bits to cleanly disable a DMA channel.</p>																											
[17]	RO	A	<p>Active:</p> <p>0 = there is no data in the FIFO of the channel</p> <p>1 = the FIFO of the channel has data.</p> <p>You can use this value with the Halt and Channel Enable bits to cleanly disable a DMA channel.</p>																											
[16]	R/W	L	Lock. When set, this bit enables locked transfers.																											
[15]	R/W	ITC	Terminal count interrupt mask. When cleared, this bit masks out the terminal count interrupt of the relevant channel.																											
[14]	R/W	IE	Interrupt error mask. When cleared, this bit masks out the error interrupt of the relevant channel.																											
[13:11]	R/W	FLOWCNTRL	Flow control and transfer type. This value indicates the flow controller and transfer type. The flow controller can be the DMAC, the source peripheral, or the destination peripheral. The transfer type can be memory-to-memory, memory-to-peripheral, peripheral-to-memory, or peripheral-to-peripheral.																											
[10]	-	-	Read undefined. Write as zero.																											
[9:6]	R/W	DESTPERIPHERALA	Destination peripheral. This value selects the DMA destination request peripheral. This field is ignored if the destination of the transfer is to memory.																											
[5]	-	-	Read undefined. Write as zero.																											

Bit	R/W	Symbol	Description
[4:1]	R/W	SRCPERIPHERALA	Source peripheral. This value selects the DMA source request peripheral. This field is ignored if the source of the transfer is from memory.
[0]	R/W	E	<p>Channel enable. Reading this bit indicates whether a channel is currently enabled or disabled:</p> <p>0 = channel disabled 1 = channel enabled.</p> <p>You can also determine the Channel Enable bit status by reading the DMACEnbldChns register.</p> <p>You enable a channel by setting this bit.</p> <p>You can disable a channel by clearing the Enable bit. This causes the current AHB transfer, if one is in progress, to complete, and the channel is then disabled. Any data in the channel's FIFO is lost. Restarting the channel by setting the Channel Enable bit has unpredictable effects and you must fully re-initialize the channel.</p> <p>The channel is also disabled, and the Channel Enable bit cleared, when the last LLI is reached, or if a channel error is encountered.</p> <p>If a channel has to be disabled without losing data in a channel's FIFO, you must set the Halt bit so that subsequent DMA requests are ignored. The Active bit must then be polled until it reaches 0, indicating that there is no data left in the channel's FIFO. Finally, you can clear the Channel Enable bit.</p>

Section 9. Interrupt Controller

9.1 Overview

64 interrupt sources from internal peripherals, including the DMA controller, UART, I2C and GPIO, etc. supply requests to an Interrupt Controller.

An FIQ or an IRQ (interrupt requests) are signaled by the Interrupt Controller to CPU. After arbitrating multiple requests from internal peripherals and GPIO, the Controller requests an interrupt.

The hardware arbitration logic decides the interrupt arbitration process and the results are recorded in the interrupt pending registers.

9.1.1 Features

The interrupt controller is responsible for:

- support for 64 IRQ interrupts
 - 2 channels, each channel processes 32 IRQ interrupts and has each AHB bus.
 - IRQ and FIQ generation
 - AHB mapped for faster interrupt response
 - software interrupt generation
 - raw interrupt status
 - interrupt request status

9.1.2 Block Diagram

The interrupt controller has two channels. Each channel has 32 interrupt sources. The interrupt request logic receives the interrupt requests from the peripheral and combines them with the software interrupt requests. It then masks out the interrupt requests which are not enabled, and routes the enabled interrupt requests to either IRQStatus or FIQStatus. Figure 9-1 shows a block diagram of the interrupt request logic in each channel.

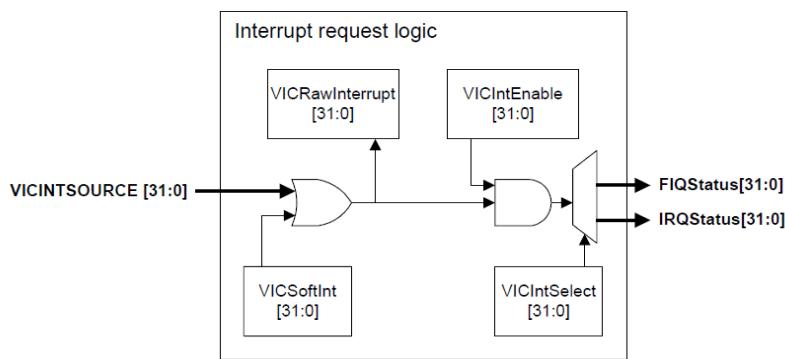


Figure 9-1. Interrupt request logic in 1 channel

The FIQ interrupt logic generates the FIQ interrupt signal by FIQ interrupt requests in the interrupt controller. Figure 9-2 shows a block diagram of the FIQ interrupt logic in each channel.

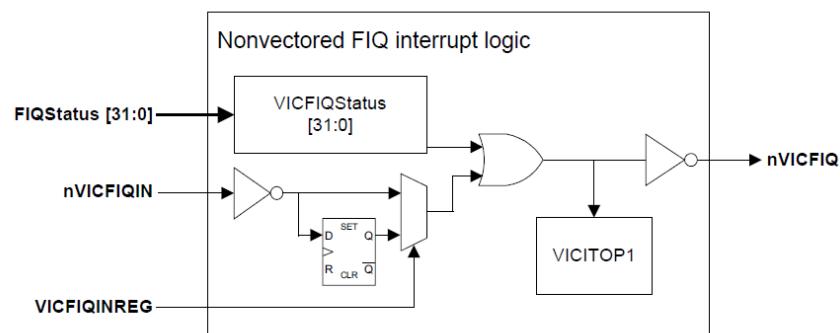


Figure 9-2. FIQ interrupt logic in 1 channel

9.2 Programming sequence

9.2.1 Interrupt flow sequence using AHB

The following procedure shows the sequence for the IRQ interrupt flow:

- An IRQ interrupt occurs.
- The ARM processor branches to the IRQ interrupt vector
- Stack the workspace so that IRQ interrupts can be re-enabled later.
- Perform a dummy read to the VICADDRESS Register to set up priority status control in the VIC.
- Read the VICIRQSTATUS Register and determine which interrupt sources have to be service.
- Execute the ISR. At the beginning of the ISR, the interrupt of the processor can be re-enabled so that a higher priority interrupt can be serviced.
- Clear the requesting interrupt in the peripheral, or write to the VICSOFTINTCLEAR Register if the request was generated by a software interrupt.
- Disable the interrupt on the processor and restore the workspace.
- Write to the VICADDRESS Register. This clears the respective interrupt in the internal interrupt priority hardware.
- Return from the interrupt. This re-enables the interrupts.

9.2.2 FIQ interrupt flow sequence

The following procedure shows the sequence for the FIQ interrupt flow.

- An FIQ interrupt occurs.
- The ARM processor branches to the FIQ interrupt vector.
- Branch to the ISR.
- Execute the ISR.
- Clear the requesting interrupt in the peripheral, or write to the VICSOFTINTCLEAR Register if the request was generated by a software interrupt.
- Disable the interrupts and restore the workspace.
- Return from the interrupt. This re-enables the interrupts

9.3 Interrupt Source

Interrupt Number	Source	Description
0	MCUSTOP	MCUSTOP interrupt
1	DMA0	DMA0 interrupt
2	DMA1	DMA1 interrupt
3	CLKPWR0	CLKPWR PWR interrupt
4	CLKPWR1	CLKPWR ALIVE interrupt
5	CLKPWR2	CLKPWR RTC interrupt
6	UART1	UART1 interrupt
7	UART0	UART0 interrupt
8	UART2	UART2 interrupt
9	UART3	UART3 interrupt
10	UART4	UART4 interrupt
11	UART5	UART5 interrupt
12	SSP0	SSP0 interrupt
13	SSP1	SSP1 interrupt
14	SSP2	SSP2 interrupt
15	I2C0	I2C0 interrupt
16	I2C1	I2C1 interrupt
17	I2C2	I2C2 interrupt
18	DEINTERLACE	DEINTERLACE interrupt
19	SCALER	SCALER interrupt
20	AC97	AC97 interrupt
21	SPDIFRX	SPDIFRX interrupt
22	SPDIFTX	SPDIFTX interrupt
23	TIMER0	TIMER0 interrupt
24	TIMER1	TIMER1 interrupt
25	TIMER2	TIMER2 interrupt
26	TIMER3	TIMER3 interrupt
27	PWM0	PWM0 interrupt
28	PWM1	PWM1 interrupt
29	PWM2	PWM2 interrupt
30	PWM3	PWM3 interrupt
31	WDT	WDT interrupt

Interrupt Number	Source	Description
32	MPEGTI	MPEGTI interrupt
33	DISPLAYTOP0	DISPLAY DUAL DISPLAY PRIM interrupt
34	DISPLAYTOP1	DISPLAY DUAL DISPLAY SECOND interrupt
35	DISPLAYTOP2	DISPLAY RESCONV interrupt
36	DISPLAYTOP3	DISPLAY HDMI interrupt
37	VIP0	VIP0 interrupt
38	VIP1	VIP1 interrupt
39	MIPI	MIPI interrupt
40	MALI400	MALI400 interrupt
41	ADC	ADC interrupt
42	PPM	PPM interrupt
43	SDMMC0	SDMMC0 interrupt
44	SDMMC1	SDMMC1 interrupt
45	SDMMC2	SDMMC2 interrupt
46	CODA9600	CODA960 HOST interrupt
47	CODA9601	CODA960 JPG interrupt
48	GMAC	GMAC interrupt
49	USB20OTG	USB20OTG interrupt
50	USB20HOST	USB20HOST interrupt
51	CAN0	CAN0 interrupt
52	CAN1	CAN1 interrupt
53	GPIOA	GPIOA interrupt
54	GPIOB	GPIOB interrupt
55	GPIOC	GPIOC interrupt
56	GPIOD	GPIOD interrupt
57	GPIOE	GPIOE interrupt
58	CRYPTO	CRYPTO interrupt
59	PDM	PDM interrupt
60	N/A	N/A
61	N/A	N/A
62	N/A	N/A
63	N/A	N/A

Table 9-1. Interrupt Sources Description Table

9.4 Register Summary

Channel0 programs 0~31 IRQ.

Channel1 programs 32~63 IRQ.

Bit	R/W	Symbol	Description	Reset Value
VICIRQSTATUS				
<i>Address : CH0 0xC0002000 / CH1 0xC0003000</i>				
[31:0]	R	VICIRQSTATUS	<p>Shows the status of the interrupts after masking by the VICINTENABLE and VICINTSELECT Registers:</p> <p>0 = interrupt is inactive (reset) 1 = interrupt is active. There is one bit of the register for each interrupt source.</p>	32h0000_0000
VICFIQSTATUS				
<i>Address : CH0 0xC0002004 / CH1 0xC0003004</i>				
[31:0]	R	VICFIQSTATUS	<p>Shows the status of the FIQ interrupts after masking by the VICINTENABLE and VICINTSELECT Registers:</p> <p>0 = interrupt is inactive (reset) 1 = interrupt is active. There is one bit of the register for each interrupt source.</p>	32h0000_0000
VICRAWINTR				
<i>Address : CH0 0xC0002008 / CH1 0xC0003008</i>				
[31:0]	R	VICRAWINTR	<p>Shows the status of the interrupts before masking by the Enable Registers:</p> <p>0 = interrupt is inactive before masking 1 = interrupt is active before masking. Because this register provides a direct view of the raw interrupt inputs, the reset value is unknown. There is one bit of the register for each interrupt source.</p>	-
VICINTSELECT				
<i>Address : CH0 0xC000200C / CH1 0xC000300C</i>				
[31:0]	R/W	VICINTSELECT	<p>Selects type of interrupt for interrupt request:</p> <p>0 = IRQ interrupt (reset) 1 = FIQ interrupt. There is one bit of the register for each interrupt source</p>	32h0000_0000
VICINTENABLE				
<i>Address : CH0 0xC0002010 / CH1 0xC0003010</i>				
[31:0]	R/W	VICINTENABLE	<p>Enables the interrupt request lines, which allow the interrupts to reach the processor.</p> <p>Read:</p> <p>0 = interrupt disabled (reset) 1 = interrupt enabled. The interrupt enable can only be set using this register. The VICINTCLEAR Register must be used to disable the interrupt enable.</p> <p>Write:</p> <p>0 = no effect 1 = interrupt enabled. On reset, all interrupts are disabled. There is one bit of the register for each interrupt source.</p>	32h0000_0000

Bit	R/W	Symbol	Description	Reset Value
VICINTCLEAR Address : CH0 0xC0002014 / CH1 0xC0003014				
[31:0]	W	VICINTCLEAR	<p>Clears corresponding bits in the VICINTENABLE Register:</p> <p>0 = no effect 1 = interrupt disabled in VICINTENABLE Register.</p> <p>There is one bit of the register for each interrupt source</p>	-
VICSOFTINT Address : CH0 0xC0002018 / CH1 0xC0003018				
[31:0]	R/W	VICSOFTINT	<p>Setting a bit HIGH generates a software interrupt for the selected source before interrupt masking.</p> <p>Read:</p> <p>0 = software interrupt inactive (reset) 1 = software interrupt active.</p> <p>Write:</p> <p>0 = no effect 1 = software interrupt enabled.</p> <p>There is one bit of the register for each interrupt source.</p>	32'h0000_0000
VICSOFTINTCLEAR Address : CH0 0xC000201C / CH1 0xC000301C				
[31:0]	W	VICSOFTINTCLEAR	<p>Clears corresponding bits in the VICSOFTINT Register:</p> <p>0 = no effect 1 = software interrupt disabled in the VICSOFTINT Register.</p> <p>There is one bit of the register for each interrupt source.</p>	-
VICPROTECTION Address : CH0 0xC0002020 / CH1 0xC0003020				
[31:1]	-	RESERVED	Reserved	-
[0]	R/W	VICPROTECTION	<p>Enables or disables protected register access:</p> <p>0 = protection mode disabled (reset) 1 = protection mode enabled.</p> <p>When enabled, only privileged mode accesses (reads and writes) can access the interrupt controller registers, that is, when HPROT[1] is set HIGH for the current transfer.</p> <p>When disabled, both user mode and privileged mode can access the registers.</p> <p>This register can only be accessed in privileged mode, even when protection mode is disabled.</p>	1'b0
VICSWPRIORITYMASK Address : CH0 0xC0002024 / CH1 0xC0003024				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	VICSWPRIORITYMASK	<p>Controls software masking of the 16 interrupt priority levels:</p> <p>0 = interrupt priority level is masked 1 = interrupt priority level is not masked (reset).</p> <p>Each bit of the register is applied to each of the 16 interrupt priority levels.</p>	16'hFFFF

Section 10. Watch Dog Timer

10.1 Overview

Watchdog timer is used to resume the controller operation whenever it is disturbed by malfunction such as noise and system error/ It can be used as normal 16bit interval timer to request interrupt service. The watchdog timer generates the reset signal.

Difference in usage WDT compared with PWM timer is that WDT generates the reset signal.

10.1.1 Features

Features of WDT are:

- Normal interval timer mode with interrupt request
- Internal reset signal is activated when the timer count value reaches 0 (time-out).
- Level-triggered interrupt mechanism

10.2 Functional Description

10.2.1 Watchdog Timer Operation

Feature 1-1 shows the functional block diagram of the watchdog timer. The watchdog timer uses only PCLK as its source clock. The PCLK frequency is prescaled to generate the corresponding watchdog timer clock, and the resulting frequency is divided again.

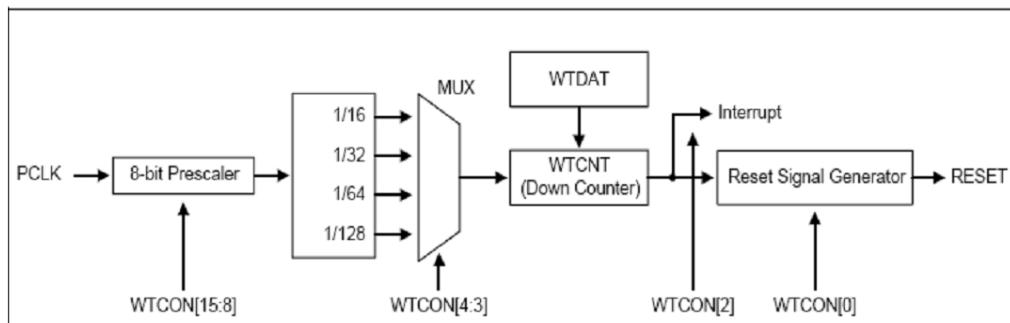


Figure 10-1. Watchdog Timer Block Diagram

The prescaler value and the frequency division factor are specified in the watchdog timer control (WDTCON) register. Valid prescaler values range from 0 to 2^8 -1. The frequency division factor can be selected as 16, 32, 64 or 128.

Use the following equation to calculate the watchdog timer clock frequency and the duration of each timer clock cycle :

$$t_{\text{watchdog}} = 1 / (\text{PCLK} / (\text{Prescaler value} + 1) / \text{Division_factor})$$

10.2.2 WTDAT & WTCNT

Once the watchdog timer is enabled, the value of watchdog timer data (WTDAT) register cannot be automatically reloaded into the timer counter (WTCNT). In this reason, an initial value must be written to the watchdog timer count (WTCNT) register, before the watchdog timer starts.

10.2.3 Consideration of Debugging Environment

When the MDIRAC-III is in debug mode Embedded ICE, the watchdog timer must not operate. The watchdog timer can determine whether or not it is currently in the debug mode from the CPU core signal (DBGACK signal). Once the DBGACK signal is asserted, the reset output of the watchdog timer is not activated as the watchdog timer is expired.

10.2.4 Special Function Register

10.2.4.1 Memory map

Register	R/W	Description	Reset Value
WDTCON	R/W	Watchdog timer control register	0x8021
WTDAT	R/W	Watchdog timer data register	0x8000

WTCNT	R/W	Watchdog timer count register	0x8000
WTCLRINT	W	Watchdog timer interrupt register	-

10.2.4.2 Watchdog timer control(WTCON) register

The WTCON register allows the user to enable/disable the watchdog timer, select the clock signal from 4 different sources, enable/disable interrupts, and enable/disable the watchdog timer output.

The Watchdog timer is used to resume restart in mal-function after its power on, if controller restart is not desired, the Watchdog timer should be disabled.

If the user wants to use the normal timer provided by the Watchdog timer, enable the interrupt and disable the Watchdog timer.

10.2.4.3 Watchdog timer data(WTDAT) register

The WTDAT register is used to specify the time-out duration. The content of WTDAT cannot be automatically loaded into the timer counter at initial watchdog timer operation. However, using 0x8000(initial value) will drive the first time_out. In this case, the value of WTDAT will be automatically reloaded into WTCNT.

10.2.4.4 Watchdog timer count(WTCNT) register

The WTCNT register contains the current count values for the watchdog timer during normal operation. Note that the content of WTDAT register cannot be automatically loaded into the timer count register when the watchdog timer is enabled initially, so the WTCNT register must be set to initial value before enabling it.

10.2.4.5 Watchdog timer interrupt(WTCLRINT) register

The WTCLRINT register is used to clear the interrupt. Interrupt service routine is responsible for clearing the relevant interrupt after the interrupt service is completed. Writing any values on this register clears the interrupt. Reading this register is not allowed.

10.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
Control register (WTCON)				
<i>Address : C001_9000h</i>				
[31:16]	-	RESERVED	Reserved	-
[15:8]	R/W	PRESCALER VALUE	Prescaler value. The valid range is from 0 to (2^8 -1).	
[7:6]	-	RESERVED	Reserved	-
[5]	R/W	WATCHDOG TIMER	Enable or disable bit of Watchdog timer. 0 = Disable 1 = Enable	
[4:3]	R/W	CLOCK SELECT	Determine the clock division factor. 00 = 16 01 = 32 10 = 64 11 = 128	
[2]	R/W	INTERRUPT GENERATION	Enable or disable bit of the interrupt. 0 = Disable 1 = Enable	
[1]	-	RESERVED	Reserved	-
[0]	R/W	RESET ENABLE/DISABLE	Enable or disable bit of Watchdog timer output for reset signal. 1 = Assert reset signal of the NXP4330D/Q at watchdog time-out. 0 = Disable the reset function of the watchdog timer.	1'h0
Data register (WTDAT)				
<i>Address : C001_9004h</i>				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	WTDAT	Watchdog timer count value for reload.	0x8000
Count register (WTCNT)				
<i>Address : C001_9008h</i>				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	WTCNT	The current value of the watchdog timer	0x8000
Interrupt register (WTCLRINT)				
<i>Address : C001_900Ch</i>				
[31:0]	W	WTCLRINT	Write any values clears the interrupt	-

Section 11. RTC

11.1 Overview

The Real Time Clock (RTC) block can be operated by the Backup Battery while the system power is off. The RTC block is composed of 32bit free counter register and works with an external 32.768 KHz Crystal and also can perform the alarm function.

11.1.1 Features

- 32bit Counter
- Alarm Function : Alarm Interrupt or Wake Up from Power Down Mode
- Independent power pin (VDD_RTC)
- Supports 1Hz Time interrupt for Power Down Mode
- Generates Power Management Reset signal

11.1.2 Block Diagram

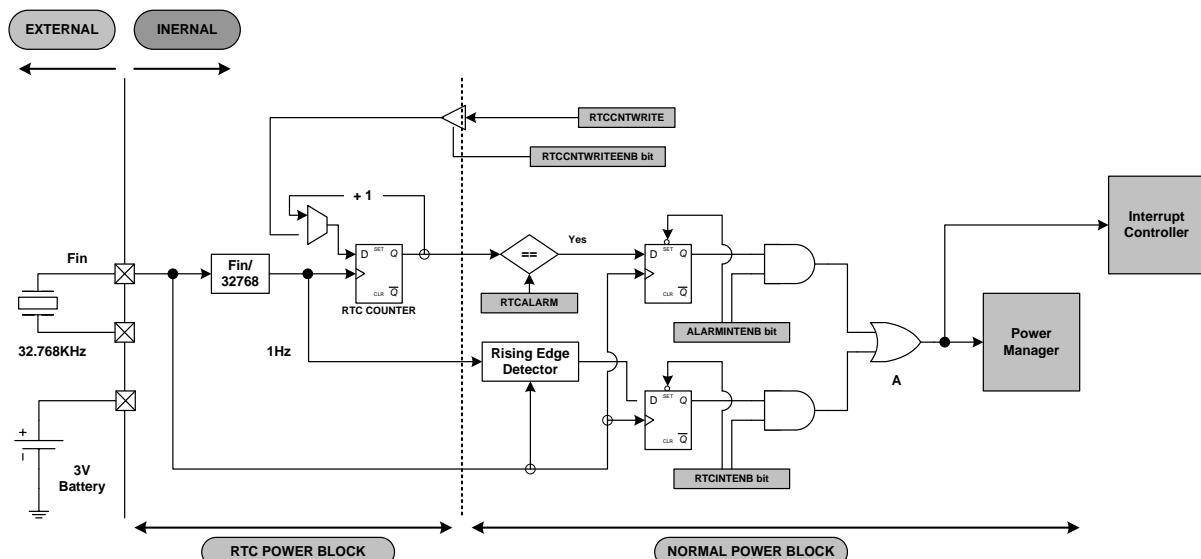


Figure 11-1. RTC Block Diagram

Figure 11-1 shows the RTC block diagram. The RTC block receives an external clock of 32.768 KHz and divides it into 1 Hz with 32.768 KHz. The RTC Counter operates depending on the external clock.

<Note 1>

As shown in Figure 11-1, the left and right parts of the central dotted line use RTC Power and Normal Power, separately. The RTC Power Block uses a mercury battery, but the block actually using the mercury battery is the RTC Counter in the RTC Power Block. The battery life is about five years.

In Figure 11-1, the output in point [A] is applied to the Power Manager or the Interrupt Controller. The output is applied to the Interrupt Controller in Normal mode and applied to the Power Manager in Power mode.

<Note 2>

Even if RTC is not used, RTC power and RTC clock should be supplied. .

11.2 Functional Description

11.2.1 Backup Battery Operation

As shown in Figure 11-1, since the RTC block uses a separate power source (Coin Battery), the RTC block operates even when the external power is turned off.

The RTC Logic can be driven by the Backup Battery, which supplies the power through the VDD_RTC pin into the RTC Block, even if the system power is off. When the system power is off, the interfaces of the CPU and RTC logic should be blocked and the backup battery only drives the oscillation circuit and the internal 32bit RTC counter to minimize power dissipation. In other words, the RTC block can be used as the Wake Up Source when the NXP4330D/Q is converted into Power Down mode.

The use of the RTC block as a Wake-Up source requires that the ***RTCCTRL.ACCESSENB*** bit is set as '0' before the system enters Power Down mode. Setting it as '0' is performed to use the RTC as the Wake-Up source even when the system enters the Power Down Mode. (For detailed information on the Power Down Mode, refer to Section 4.)

Even if the RTC block is not used, the RTC Clock must be connected to NXP4330D/Q because the RTC clock is used as the clock for power management operation.

11.2.2 RTC Operation

The RTC generates an alarm signal at a specified time in the Power Down Mode or Normal Operation Mode. In Normal Operation Mode, the Alarm Interrupt is activated. In Power Down Mode, the Power Management Wake Up Signal is activated as well as the ***RTCALARM***. The ALARM Time Set register (***RTCALARM***) determines the condition of the alarm time setting and the ***RTCINTENB.ALARMINTENB*** bit determines the alarm enable/disable status.

The procedure to generate an alarm interrupt is as follows:

First, write a counter value to the ***RTCCNTWRITE*** register. (To this end, the busy status of the ***RTCCTRL.RTCCNTWAIT*** should be checked in advance. The written value is applied to the register after two 32.768 KHz clock cycles.) After that, write the value of the point at which you wish to generate an interrupt to the ***RTCALARM*** register. The RTC counter increases the counter value at intervals of 1 Hz. If the values of the two registers (***RTCCNTWRITE*** and ***RTCAKARM*** registers) become equal when the ***RTCINTENB.ALARMINTENB*** bit is set as '1', an interrupt occurs.

In a similar way, the RTC interrupt is detected in a rising edge of 1Hz. In this case, the interrupt is generated by setting the ***RTCINTENB.RTCINTENB*** bit as '1'.

In addition, The ***RTCINTENB*** register contains the Pending Clear function and the Pending Clear is performed by writing '0'.

11.2.3 Accessing the RTC Time Counter Setting/Read Register

To access RTC Time Count Read Register (RTCCNTREAD) and RTC Time Count Setting Register (RTCCNTWRITE), the ***RTCCTRL.RTCCNTWRITEENB*** bit is set to '1' before accessing these register. When the CPU completes to access these register, the CPU should set the ***RTCCTRL.RTCCNTWRITEENB*** bit to '0' to protect the content of RTC counter from unknown problem in abnormal state. The ***RTCCNTWRITEENB*** bit determines the reflection of the RCCNTWRITE register value to the RTC counter.

11.2.4 Interrupt Pending Register

Only the “READ” function is available for the **RTCINTPND** register of the NXP4330D/Q, but the current pending status can be read.

Since the **RTCINTPND** register only has a “READ” function, the Pending Clear function is controlled by the **RTCINTENB** register. The Interrupt Pending status is cleared by disabling the relevant interrupt. Therefore, if the corresponding bit of the **RTCINTENB** register is set as ‘1’, the relevant interrupt is enabled. If the corresponding bit is set as ‘0’, the interrupt is disabled and the pending bit is also cleared.

11.2.5 Power Manager Reset Time Control

RTC controls the time when nPWRMANRST (Power Manager Reset) releases from CorePOR. Refer to RTCCORERSTIMESEL Register for setting the time when nPWRMANRST releases.

11.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
			WRITE> 0: Pending Clear & Interrupt Disable 1 : Interrupt Enable	
[0]	R/W	RTCINTENB	RTCINTENB : Set RTC(1Hz Only) Interrupt On/Off and Pending Clear/ Interrupt Enable READ> 0: Interrupt Disable 1 : Interrupt Enable WRITE> 0: Pending Clear & Interrupt Disable 1 : Interrupt Enable	1'b0

RTC INTERRUPT PENDING REGISTER (RTCINTPND)

Address : C001_0C14h

[31 : 2]	-	RESERVED	Reserved	30b0
[1]	R	ALARMINTPEND	ALARMINTPEND : ALARM interrupt Pending bit. 0: None 1 : Interrupt Pended	1'b0
[0]	R	RTCINTPEND	RTCINTPEND : Set RTC (1Hz Only) interrupt Pending bit. 0: None 1 : Interrupt Pended	1'b0

RTC CORE POR TIME SELECT REGISTER (RTCCORERSTIMESEL)

Address : C001_0C18h

[31 : 2]	-	RESERVED	Reserved	30b0
[6:0]	R/W	COREPORTIMESEL6	CORE POR (Power On Reset) releases with the delay of about 186ms after CORE VDD Power is turned on. 0: None 1: Select	1'b0
[5]	R/W	COREPORTIMESEL5	CORE POR (Power On Reset) releases with the delay of about 155ms after CORE VDD Power is turned on.. 0: None 1: Select	1'b0
[4]	R/W	COREPORTIMESEL4	CORE POR (Power On Reset) releases with the delay of about 124ms after CORE VDD Power is turned on.. 0: None 1: Select	1'b0
[3]	R/W	COREPORTIMESEL3	CORE POR (Power On Reset) releases with the delay of about 93ms after CORE VDD Power is turned on.. 0: None 1: Select	1'b0
[2]	R/W	COREPORTIMESEL2	CORE POR (Power On Reset) releases with the delay of about 62ms after CORE VDD Power is turned on.. 0: None 1: Select	1'b0
[1]	R/W	COREPORTIMESEL1	CORE POR (Power On Reset) releases with the delay of about 31ms after CORE VDD Power is turned on.. 0: None 1: Select	1'b0
[0]	R/W	COREPORTIMESEL0	CORE POR (Power On Reset) releases without delay after CORE VDD Power is turned on. 0: None 1: Select	1'b0

<note> CORE POR releases about 210ms after CORE VDD Power is turned on when RTCCOREPORTIMESEL[6:0] == 7b0

RTC SCRATCH REGISTER (RTCSCRATCH)

Address : C001_0C1Ch

[31:0]	R/W	RTCSCRATCH	RTC Scratch register.	32b0
--------	-----	------------	-----------------------	------

Section 12. Alive

12.1 Overview

In the status with eliminating core power supply of NXP4330D/Q, some PAD need power supply continuously and should keep driving PAD with certain value.

For example a bit that controlling STN LCD should keep driving PAD with low in the status with eliminating core power supply. 32bit value can be saved in Scratch Register and the saved value maintains in the case of core power off.

User could on/off the system power by pressing toggle switch and ALIVE performs the necessary functions in these momentary power controls.

12.1.1 Features

- AliveGPIO PADs are all in/output PAD.
- The value of ALIVE Block maintains even in power off of Core Power.
- Alive Block does not use clock. To change the control register value, need to program set/reset pin of SR-flipflop directly.
- Chip sleep mode wake-up source(AliveGPIO, VDDToggle, RTCIRQ)
- Power IC Enable.
- Supports the PAD Hold function

Scan chain is not inserted to Alive GPIO.

12.2 Power Isolation

12.2.1 Core Power Off

In the case of power off of CoreVDD, Alive Registers maintains its value since Alive VDD keep supplied. Pull-down register connected to nPowerGating performs the function of maintaining control bit of Alive Registers securely in the interval of core power off or unstable.

12.2.2 Power Gating

The value of set/reset should be kept as low to maintains the value of Alive Registers securely in the case of power off of CoreVDD. Therefore, it's designed to maintain low value in the case of core power off since nPowerGating register is connected to pull-down register.

12.3 Alive Registers

All of the Registers except nPowerGating/AliveDetectPending Register of Alive block maintains the value written in Register when Core power turns off, and are reset when Alive Power turns off. Alive Registers do not have their own Clock, and these can be read and written when nPowerGating register is '1' (nPowerGating='1'). And especially, in write mode, Alive Registers can be written by SET/RST (reset) Register. The following is the example of the Register function according to Register SET/RST.

Ex) Alive Registers remains as the former state when Regiser_nameSET == 0, Regiser_nameRST=0

Alive Registers are written as '1' when Regiser_nameSET == 1, Regiser_nameRST=0

Alive Registers are written as '0' write '0' when Regiser_nameSET == 0, Regiser_nameRST=1

Alive Registers are written as '1' when Regiser_nameSET == 1, Regiser_nameRST=1

12.3.1 Alive GPIO Detect Registers

For Alive GPIO input, Alive GPIO Detect Registers can be used as Core Power on, Alive Interrupt, Sleep mode wakeup source in case of Asynchronous/synchronous detecting mode. And when those events are detected, ALIVEGPIODETECTPENDREG[n] Register is set to '1'. The following are the operating examples according to Detecting modes.

Ex1) Asynchronous Low Level Detecting

Alive GPIO Detect Registers detect the event when ALIVEGPIOLOWASYNCDETECTMODEREADREG [n] = '1', ALIVEGPIODETECTENBREADREG[n] = '1', and AliveGPIO =='0'

Ex2) Asynchronous High Level Detecting

Alive GPIO Detect Registers detect the event when ALIVEGPIOHIGHASYNCDETECTMODEREADREG [n] = '1', ALIVEGPIODETECTENBREADREG[n] = '1', and AliveGPIO[n] == '1'

Ex3) Synchronous Low Level Detecting

Alive GPIO Detect Registers detect the event when ALIVEGPIOLOWSYNCDETECTMODEREADREG [n] = '1', ALIVEGPIODETECTENBREADREG[n] = '1', AliveGPIO[n] == '0'

Ex4) Synchronous Hight Level Detecting

Alive GPIO Detect Registers detect the event when ALIVEGPIOHIGHSYNCDETECTMODEREADREG [n] = '1', ALIVEGPIODETECTENBREADREG[n] = '1', AliveGPIO[n] == '1'

Ex5) Synchronous Falling Edge Detecting

Alive GPIO Detect Registers detect the event when ALIVEGPIOFALLDETECTMODEREADREG[n] = '1', ALIVEGPIODETECTENBREADREG[n] = '1', and AliveGPIO[n] bit changes from '1' to '0'

Ex6) Synchronous Rising Edge Detecting

Alive GPIO Detect Registers detect the event when ALIVEGPIORISEDETECTMODEREADREG [n] = '1', ALIVEGPIODETECTENBREADREG[n] = '1', and AliveGPIO[n] bit changes from '0' to '1'

12.3.2 Scratch Register

Programmer could save any 32bit value in Scratch Register. The value of Scratch Register maintains in the case of power off of CoreVDD.

12.3.3 Alive GPIO Control Registers

Alive GPIO is controlled through Alive Block regardless of GPIO block.

Control Register has AliveGPIO in/out mode enable, pull-up, and AliveGPIOPADOut.

12.4 Momentary Power Control

12.4.1 CoreVDD Powering on

The Core Power can be changed from off-state to on-state by the VDDPWRToggle switch, AliveGPIO Detecting, RTC interrupt. And in case of Power on, VDDPWRToggle switch, AliveGPIO Detecting, and RTC interrupt can be processed, after system booting, by setting VDDPWRON_DDR/VDDPWRON Bit. (Power On is not allowed when Battery Fault occurs.).

12.4.2 CoreVDD Powering off

The Core Power can be changed from on-state to off-state by clearing VDDPWRON_DDR/VDDPWRON Bit.

Core Block and Alive Block are connected through PowerGating Register, which makes it possible for Alive Registers to safely sustain their own values even after Core Power turns off with VDDPWRON_DDR/VDDPWRON Bit cleared.

The following is the example of Chip Power sequence

- 1) Do not hold the initial Pad state. (*nPadHold*=1)
- 2) Hold the Pad before you turn off the Power. (*nPadHold* Register=0)
- 3) Turn off VDDPWRON
- 4) Non-Alive POR is set low after power turns off
- 5) Power starts to be supplied by pressing toggle switch
- 6) PAD releases from the hold state when internal POR turns on (which provides the stability for preventing pad hold from releasing after por turns on)
- 7) VDDPWRON and PadHold Register should be released simultaneously after system booting.
- 8) VDDPOWRON should be released.

nPadHoldEnb Register should be set as '0' except Power On Reset.

12.5 SLEEP Mode

NXP4330D/Q supports two SLEEP Modes(SleepMode1, SleepMode2). Core power turns off in SLEEP Mode, and the Chip wakes up from SLEEP Mode by Wake-up sources such as AliveGPIO Detecting, RTC Interrupt, and nVDDPWRTOGGLE Switch push.(And, No wake-up is possible in case of battery fault)

- Alive GPIO Detect Pending Register Clear
- Hold the Pad before user turns off Power. (SleepMode1 : nPadHold[2:1]=2'b00, SleepMode2 : nPadHold[2:0]== 3'b000)
- VDDPWRON Register Clear
- nPowerGating Register Clear
- NXP4330D/Q STOP Mode Set(Refer to Clock and Power management Section)

12.6 PMU (Power management Unit)

12.6.1 Overview

PMU is a block inside of ALIVE. It controls internal power switch of sub-blocks in chip. PMU can controls ther power up and down of these blocks:

- GPU (graphic processing unit)
- MFC (multi function codec)

12.6.2 Power Mode Table

MODE		PD_RTC	PD_ALIVE	PD_TOP	PD_DREX	PD_CODEC	PD_GR3D	PD_CPU
Num	Name							
1	POWER ON RESET	ON	ON	ON	ON	ON	ON	ON
2	NORMAL	ON	ON	ON	ON	ON	ON	ON
3	SUB SLEEP - 0	ON	ON	ON	ON	OFF	ON	ON
4	SUB SLEEP - 1	ON	ON	ON	ON	ON	OFF	ON
5	SUB SLEEP - ALL	ON	ON	ON	ON	OFF	OFF	ON
6	DEEP IDLE	ON	ON	ON	ON	OFF	OFF	OFF
7	TOP SLEEP	ON	ON	OFF	OFF	OFF	OFF	OFF
8	RTC ONLY	ON	OFF	OFF	OFF	OFF	OFF	OFF

12.6.3 Switch Control Sequence

12.6.3.1 Power Off Sequence

1. Set '0' on PMUNISOLATE register
2. Set '0' on PMUPWRUPPRE register
3. Set '0' on PMUPWRUP register
4. Wait until PMUPWRUPACK signal goes '0'

12.6.3.2 Power On Sequence

1. Set '1' on PMUPWRUP register
2. Set '1' on PMUPWRUPPRE register
3. Wait until PMUPWRUPACK signal goes '1'
4. Set '1' on PMUNISOLATE register

12.7 Register Summary

Bit	R/W	Symbol	Description	Reset Value
ALIVE POWER GATING REGISTER (ALIVEPWRGATEREG)				
<i>Address : C001_0800h</i>				
[31:1]	-	RESERVED	Reserved	0
[0]	R/W	NPOWERGATING	nPowerGating (negative active Power Gating). The default value is 0, disabling writing to Alive Registers, in order to keep the values of Alive Registers when Core Power 1.0V is off. 0 : Disable writing data to Alive Register 1 : Enable writing data to Alive Register	1b0
Alive GPIO ASync Detect Mode Reset Register 0(ALIVEGPIOASYNCDETECTMODERSTREG0)				
<i>Address : C001_0804h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ASYNCDETECTMODERS_T0_7	Alive GPIO7 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
[6]	R/W	ASYNCDETECTMODERS_T0_6	Alive GPIO6 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
[5]	R/W	ASYNCDETECTMODERS_T0_5	Alive GPIO5 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
[4]	R/W	ASYNCDETECTMODERS_T0_4	Alive GPIO4 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
[3]	R/W	ASYNCDETECTMODERS_T0_3	Alive GPIO3 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
[2]	R/W	ASYNCDETECTMODERS_T0_2	Alive GPIO2 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
[1]	R/W	ASYNCDETECTMODERS_T0_1	Alive GPIO1 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
[0]	R/W	ASYNCDETECTMODERS_T0_0	Alive GPIO0 Low Level Async detect mode Register Reset. 0 : none 1: Reset	1b0
Alive GPIO ASync Detect Mode Set Register 0 (ALIVEGPIOASYNCDETECTMODESETREG0)				
<i>Address : C001_0808h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ASYNCDETECTMODESE_T0_7	Alive GPIO7 Low Level Async detect mode Register Set. 0 : none 1: Set	1b0
[6]	R/W	ASYNCDETECTMODESE_T0_6	Alive GPIO6 Low Level Async detect mode Register Set. 0 : none 1: Set	1b0
[5]	R/W	ASYNCDETECTMODESE_T0_5	Alive GPIO5 Low Level Async detect mode Register Set. 0 : none 1: Set	1b0
[4]	R/W	ASYNCDETECTMODESE_T0_4	Alive GPIO4 Low Level Async detect mode Register Set. 0 : none 1: Set	1b0
[3]	R/W	ASYNCDETECTMODESE_T0_3	Alive GPIO3 Low Level Async detect mode Register Set. 0 : none 1: Set	1b0
[2]	R/W	ASYNCDETECTMODESE_T0_2	Alive GPIO2 Low Level Async detect mode Register Set.	1b0

Bit	R/W	Symbol	Description	Reset Value
		T0_2	0 : none 1 : Set	
[1]	R/W	ASYNCDETECTMODESET0_1	Alive GPIO1 Low Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[0]	R/W	ASYNCDETECTMODESET0_0	Alive GPIO0 Low Level Async detect mode Register Set. 0 : none 1 : Set	1b0
ALIVE GPIO Low Level Async Detect Mode Read Register (ALIVEGPIOLOWASYNCDETECTMODEREADREG)				
<i>Address : C001_080Ch</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOLOWASYNCDETCTMODE7	Alive GPIO7 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIOLOWASYNCDETCTMODE6	Alive GPIO6 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIOLOWASYNCDETCTMODE5	Alive GPIO5 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIOLOWASYNCDETCTMODE4	Alive GPIO4 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIOLOWASYNCDETCTMODE3	Alive GPIO3 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIOLOWASYNCDETCTMODE2	Alive GPIO2 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIOLOWASYNCDETCTMODE1	Alive GPIO1 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIOLOWASYNCDETCTMODE0	Alive GPIO0 Low level Async detect mode register 0 : Disable 1 : Enable	1b0
<Note> ALIVEGPIOLOWASYNCDETECT(n) Register operates as follows				
It remains as the former state in case of {ASYNCDETECTENBRST0_(n), ASYNCDETECTENBSET0_(n)} = 2'b00				
It is set as '1' in case of {ASYNCDETECTENBRST0_(n), ASYNCDETECTENBSET0_(n)} = 2'b01				
It is set as '0' in case of {ASYNCDETECTENBRST0_(n), ASYNCDETECTENBSET0_(n)} = 2'b10				
It is set as '1' in case of {ASYNCDETECTENBRST0_(n), ASYNCDETECTENBSET0_(n)} = 2'b11				
Alive GPIO ASync Detect Mode Reset Register 1 (ALIVEGPIOASYNCDETECTMODERSTREG1)				
<i>Address : C001_0810h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ASYNCDETECTMODERS T1_7	Alive GPIO7 High Level Async detect mode Register Reset. 0 : none 1 : Reset	1b0
[6]	R/W	ASYNCDETECTMODERS T1_6	Alive GPIO6 High Level Async detect mode Register Reset. 0 : none 1 : Reset	1b0
[5]	R/W	ASYNCDETECTMODERS T1_5	Alive GPIO5 High Level Async detect mode Register Reset. 0 : none 1 : Reset	1b0
[4]	R/W	ASYNCDETECTMODERS T1_4	Alive GPIO4 High Level Async detect mode Register Reset. 0 : none 1 : Reset	1b0
[3]	R/W	ASYNCDETECTMODERS T1_3	Alive GPIO3 High Level Async detect mode Register Reset. 0 : none 1 : Reset	1b0
[2]	R/W	ASYNCDETECTMODERS	Alive GPIO2 High Level Async detect enable Register Reset.	1b0

Bit	R/W	Symbol	Description	Reset Value
		T1_2	0 : none 1 : Reset	
[1]	R/W	ASYNCDETECTMODERS T1_1	Alive GPIO1 High Level Async detect mode Register Reset. 0 : none 1 : Reset	1b0
[0]	R/W	ASYNCDETECTMODERS T1_0	Alive GPIO0 High Level Async detect mode Register Reset. 0 : none 1 : Reset	1b0

Alive GPIO ASync Detect Mode Set Register 1(ALIVEGPIOASYNCDETECTMODESETREG1)

Address : C001_0814h

[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ASYNCDETECTMODESE T1_7	Alive GPIO7 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[6]	R/W	ASYNCDETECTMODESE T1_6	Alive GPIO6 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[5]	R/W	ASYNCDETECTMODESE T1_5	Alive GPIO5 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[4]	R/W	ASYNCDETECTMODESE T1_4	Alive GPIO4 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[3]	R/W	ASYNCDETECTMODESE T1_3	Alive GPIO3 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[2]	R/W	ASYNCDETECENBSET1_2	Alive GPIO2 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[1]	R/W	ASYNCDETECTMODESE T1_1	Alive GPIO1 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0
[0]	R/W	ASYNCDETECTMODESE T1_0	Alive GPIO0 High Level Async detect mode Register Set. 0 : none 1 : Set	1b0

ALIVE GPIO High Level Async Detect Mode Read Register (ALIVEGPIOHIGHASYNCDETECTMODEREADREG)

Address : C001_0818h

[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOHIGHASYNC DETECTMODE7	Alive GPIO7 High level Async detect mode register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIOHIGHASYNC DETECTMODE6	Alive GPIO6 High level Async detect mode register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIOHIGHASYNC DETECTMODE5	Alive GPIO5 High level Async detect mode register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIOHIGHASYNC DETECTMODE4	Alive GPIO4 High level Async detect mode register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIOHIGHASYNC DETECTMODE3	Alive GPIO3 High level Async detect mode register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIOHIGHASYNC DETECTMODE2	Alive GPIO2 High level Async detect mode register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIOHIGHASYNC DETECTMODE1	Alive GPIO1 High level Async detect mode register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIOHIGHASYNC DETECTMODE0	Alive GPIO0 High level Async detect mode register 0 : Disable 1 : Enable	1b0

Bit	R/W	Symbol	Description	Reset Value
<Note> ALIVEGPIOHIGHASYNCDETECTMODE(n) Register operates as follows				
It remains as the former state in case of {ASYNCDETECTMODERST1_(n), ASYNCDETECTMODESET1_(n)} = 2'b00				
It is set as '1' in case of {ASYNCDETECTMODERST1_(n), ASYNCDETECTMODESET1_(n)} = 2'b01				
It is set as '0' in case of {ASYNCDETECTMODERST1_(n), ASYNCDETECTMODESET1_(n)} = 2'b10				
It is set as '1' in case of {ASYNCDETECTMODERST1_(n), ASYNCDETECTMODESET1_(n)} = 2'b11				
Allive GPIO Detect Mode Reset Register 0 (ALIVEGPIODETECTMODERSTREG0)				
<i>Address : C001_081Ch</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODERST0_7	Alive GPIO7 Falling Edge detect mode Register Reset. 0: none 1: Reset	1b0
[6]	R/W	DETECTMODERST0_6	Alive GPIO6 Falling Edge detect mode Register Reset. 0: none 1: Reset	1b0
[5]	R/W	DETECTMODERST0_5	Alive GPIO5 Falling Edge detect mode Register Reset. 0: none 1: Reset	1b0
[4]	R/W	DETECTMODERST0_4	Alive GPIO4 Falling Edge detect mode Register Reset. 0: none 1: Reset	1b0
[3]	R/W	DETECTMODERST0_3	Alive GPIO3 Falling Edge detect mode Register Reset. 0: none 1: Reset	1b0
[2]	R/W	DETECTMODERST0_2	Alive GPIO2 Falling Edge detect enable Register Reset 0: none 1: Reset	1b0
[1]	R/W	DETECTMODERST0_1	Alive GPIO1 Falling Edge detect mode Register Reset. 0: none 1: Reset	1b0
[0]	R/W	DETECTMODERST0_0	Alive GPIO0 Falling Edge detect mode Register Reset. 0: none 1: Reset	1b0
Allive GPIO Detect Mode Set Register 0 (ALIVEGPIODETECTMODESETREG0)				
<i>Address : C001_0820h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODESET0_7	Alive GPIO7 Falling Edge detect mode Register Set. 0: none 1: Set	1b0
[6]	R/W	DETECTMODESET0_6	Alive GPIO6 Falling Edge detect mode Register Set. 0: none 1: Set	1b0
[5]	R/W	DETECTMODESET0_5	Alive GPIO5 Falling Edge detect mode Register Set. 0: none 1: Set	1b0
[4]	R/W	DETECTMODESET0_4	Alive GPIO4 Falling Edge detect mode Register Set. 0: none 1: Set	1b0
[3]	R/W	DETECTMODESET0_3	Alive GPIO3 Falling Edge detect mode Register Set. 0: none 1: Set	1b0
[2]	R/W	DETECTMODESET0_2	Alive GPIO2 Falling Edge detect mode Register Set. 0: none 1: Set	1b0
[1]	R/W	DETECTMODESET0_1	Alive GPIO1 Falling Edge detect mode Register Set. 0: none 1: Set	1b0
[0]	R/W	DETECTMODESET0_0	Alive GPIO0 Falling Edge detect mode Register Set. 0: none 1: Set	1b0

Bit	R/W	Symbol	Description	Reset Value
ALIVE GPIO Falling Edge Detect Mode Read Register (ALIVEGPIOFALLDETECTMODEREADREG)				
<i>Address : C001_0824h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOFALLDETECT MODE7	Alive GPIO7 Falling Edge detect mode register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIOFALLDETECT MODE6	Alive GPIO6 Falling Edge detect mode register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIOFALLDETECT MODE5	Alive GPIO5 Falling Edge detect mode register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIOFALLDETECT MODE4	Alive GPIO4 Falling Edge Async detect mode register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIOFALLDETECT MODE3	Alive GPIO3 Falling Edge detect mode register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIOFALLDETECT MODE2	Alive GPIO2 Falling Edge detect mode register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIOFALLDETECT MODE1	Alive GPIO1 Falling Edge detect mode register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIOFALLDETECT MODE0	Alive GPIO0 Falling Edge detect mode register 0 : Disable 1 : Enable	1b0
<Note> ALIVEGPIOFALLDETECTMODE(n) Register operates as follows				
It remains as the former state in case of {DETECTMODERST0_(n), DETECTMODESET0_(n)} = 2'b00				
It is set as '1' in case of {DETECTMODERST0_(n), DETECTMODESET0_(n)} = 2'b01				
It is set as '0' in case of {DETECTMODERST0_(n), DETECTMODESET0_(n)} = 2'b10				
It is set as '1' in case of {DETECTMODERST0_(n), DETECTMODESET0_(n)} = 2'b11				
Alive GPIO Detect Mode Reset Register 1 (ALIVEPIODETECTMODERSTREG1)				
<i>Address : C001_0828h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODERST1_7	Alive GPIO7 Rising Edge detect mode Register Reset. 0 : none 1: Reset	1b0
[6]	R/W	DETECTMODERST1_6	Alive GPIO6 Rising Edge detect mode Register Reset. 0 : none 1: Reset	1b0
[5]	R/W	DETECTMODERST1_5	Alive GPIO5 Rising Edge detect mode Register Reset. 0 : none 1: Reset	1b0
[4]	R/W	DETECTMODERST1_4	Alive GPIO4 Rising Edge detect mode Register Reset. 0 : none 1: Reset	1b0
[3]	R/W	DETECTMODERST1_3	Alive GPIO3 Rising Edge detect mode Register Reset. 0 : none 1: Reset	1b0
[2]	R/W	DETECTMODERST1_2	Alive GPIO2 Rising Edge detect enable Register Reset. 0 : none 1: Reset	1b0
[1]	R/W	DETECTMODERST1_1	Alive GPIO1 Rising Edge detect mode Register Reset. 0 : none 1: Reset	1b0
[0]	R/W	DETECTMODERST1_0	Alive GPIO0 Rising Edge detect mode Register Reset. 0 : none 1: Reset	1b0

Bit	R/W	Symbol	Description	Reset Value
Alive GPIO Detect Mode Set Register 1 (ALIVEGPIODETECTMODESETREG1)				
<i>Address : C001_082Ch</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODESET1_7	Alive GPIO7 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
[6]	R/W	DETECTMODESET1_6	Alive GPIO6 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
[5]	R/W	DETECTMODESET1_5	Alive GPIO5 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
[4]	R/W	DETECTMODESET1_4	Alive GPIO4 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
[3]	R/W	DETECTMODESET1_3	Alive GPIO3 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
[2]	R/W	DETECTMODESET1_2	Alive GPIO2 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
[1]	R/W	DETECTMODESET1_1	Alive GPIO1 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
[0]	R/W	DETECTMODESET1_0	Alive GPIO0 Rising Edge detect mode Register Set. 0 : none 1: Set	1b0
ALIVE GPIO Rising Edge Detect Mode Read Register (ALIVEGPIORISEDETECTMODEREADREG)				
<i>Address : C001_9030h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIORISEDETECT MODE7	Alive GPIO7 Rising Edge detect mode register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIORISEDETECT MODE6	Alive GPIO6 Rising Edge detect mode register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIORISEDETECT MODE5	Alive GPIO5 Rising Edge detect mode register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIORISEDETECT MODE4	Alive GPIO4 Rising Edge Async detect mode register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIORISEDETECT MODE3	Alive GPIO3 Rising Edge detect mode register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIORISEDETECT MODE2	Alive GPIO2 Rising Edge detect mode register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIORISEDETECT MODE1	Alive GPIO1 Rising Edge detect mode register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIORISEDETECT MODE0	Alive GPIO0 Rising Edge detect mode register 0 : Disable 1 : Enable	1b0
<Note> ALIVEGPIORISEDETECTMODE(n) Register operates as follows				
It remains as the former state in case of {DETECTMODERST1_(n), DETECTMODESET1_(n)} = 2'b00				
It is set as '1' in case of {DETECTMODERST1_(n), DETECTMODESET1_(n)} = 2'b01				
It is set as '0' in case of {DETECTMODERST1_(n), DETECTMODESET1_(n)} = 2'b10				
It is set as '1' in case of {DETECTMODERST1_(n), DETECTMODESET1_(n)} = 2'b11				

Bit	R/W	Symbol	Description	Reset Value
Allive GPIO Detect Mode Reset Register 2 (ALIVEGPIODETECTMODERSTREG2)				
<i>Address : C001_0834h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODERST2_7	Alive GPIO7 Low Level detect mode Register Reset. 0 : none 1: Reset	1b0
[6]	R/W	DETECTMODERST2_6	Alive GPIO6 Low Level detect mode Register Reset. 0 : none 1: Reset	1b0
[5]	R/W	DETECTMODERST2_5	Alive GPIO5 Low Level detect mode Register Reset. 0 : none 1: Reset	1b0
[4]	R/W	DETECTMODERST2_4	Alive GPIO4 Low Level detect mode Register Reset. 0 : none 1: Reset	1b0
[3]	R/W	DETECTMODERST2_3	Alive GPIO3 Low Level detect mode Register Reset. 0 : none 1: Reset	1b0
[2]	R/W	DETECTMODERST2_2	Alive GPIO2 Low Level detect enable Register Reset. 0 : none 1: Reset	1b0
[1]	R/W	DETECTMODERST2_1	Alive GPIO1 Low Level detect mode Register Reset. 0 : none 1: Reset	1b0
[0]	R/W	DETECTMODERST2_0	Alive GPIO0 Low Level detect mode Register Reset. 0 : none 1: Reset	1b0
Allive GPIO Detect Mode Set Register 2 (ALIVEGPIODETECTMODESETREG2)				
<i>Address : C001_0838h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODESET2_7	Alive GPIO7 Low Level detect mode Register Set. 0 : none 1: Set	1b0
[6]	R/W	DETECTMODESET2_6	Alive GPIO6 Low Level detect mode Register Set. 0 : none 1: Set	1b0
[5]	R/W	DETECTMODESET2_5	Alive GPIO5 Low Level detect mode Register Set. 0 : none 1: Set	1b0
[4]	R/W	DETECTMODESET2_4	Alive GPIO4 Low Level detect mode Register Set. 0 : none 1: Set	1b0
[3]	R/W	DETECTMODESET2_3	Alive GPIO3 Low Level detect mode Register Set. 0 : none 1: Set	1b0
[2]	R/W	DETECTMODESET2_2	Alive GPIO2 Low Level detect mode Register Set. 0 : none 1: Set	1b0
[1]	R/W	DETECTMODESET2_1	Alive GPIO1 Low Level detect mode Register Set. 0 : none 1: Set	1b0
[0]	R/W	DETECTMODESET2_0	Alive GPIO0 Low Level detect mode Register Set. 0 : none 1: Set	1b0
ALIVE GPIO Low Level Detect Mode Read Register (ALIVEGPIOLOWDETECTMODEREADREG)				
<i>Address : C001_083Ch</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOLOWDETECT MODE7	Alive GPIO7 Low Level detect mode register 0 : Disable 1 : Enable	1b0

Bit	R/W	Symbol	Description	Reset Value
[6]	R	ALIVEGPIOLOWDETECT MODE6	Alive GPIO6 Low Level detect mode register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIOLOWDETECT MODE5	Alive GPIO5 Low Level detect mode register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIOLOWDETECT MODE4	Alive GPIO4 Low Level Async detect mode register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIOLOWDETECT MODE3	Alive GPIO3 Low Level detect mode register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIOLOWDETECT MODE2	Alive GPIO2 Low Level detect mode register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIOLOWDETECT MODE1	Alive GPIO1 Low Level detect mode register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIOLOWDETECT MODE0	Alive GPIO0 Low Level detect mode register 0 : Disable 1 : Enable	1b0

<Note> ALIVEGPIOLOWDETECTMODE(n) Register operates as follows

It remains as the former state in case of {DETECTMODERST2_(n), DETECTMODESET2_(n)} = 2'b00

It is set as '1' in case of {DETECTMODERST2_(n), DETECTMODESET2_(n)} = 2'b01

It is set as '0' in case of {DETECTMODERST2_(n), DETECTMODESET2_(n)} = 2'b10

It is set as '1' in case of {DETECTMODERST2_(n), DETECTMODESET2_(n)} = 2'b11

Allive GPIO Detect Mode Reset Register 3 (ALIVEGPIODETECTMODERSTREG3)

Address : C001_0840h

[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODERST3_7	Alive GPIO7 High Level detect mode Register Reset. 0 : none 1: Reset	1b0
[6]	R/W	DETECTMODERST3_6	Alive GPIO6 High Level detect mode Register Reset. 0 : none 1: Reset	1b0
[5]	R/W	DETECTMODERST3_5	Alive GPIO5 High Level detect mode Register Reset. 0 : none 1: Reset	1b0
[4]	R/W	DETECTMODERST3_4	Alive GPIO4 High Level detect mode Register Reset. 0 : none 1: Reset	1b0
[3]	R/W	DETECTMODERST3_3	Alive GPIO3 High Level detect mode Register Reset. 0 : none 1: Reset	1b0
[2]	R/W	DETECTMODERST3_2	Alive GPIO2 High Level detect enable Register Reset. 0 : none 1: Reset	1b0
[1]	R/W	DETECTMODERST3_1	Alive GPIO1 High Level detect mode Register Reset. 0 : none 1: Reset	1b0
[0]	R/W	DETECTMODERST3_0	Alive GPIO0 High Level detect mode Register Reset. 0 : none 1: Reset	1b0

Allive GPIO Detect Mode Set Register 3 (ALIVEGPIODETECTMODESETREG3)

Address : C001_0844h

[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTMODESET3_7	Alive GPIO7 High Level detect mode Register Set. 0 : none 1: Set	1b0

Bit	R/W	Symbol	Description	Reset Value
[6]	R/W	DETECTMODESET3_6	Alive GPIO6 High Level detect mode Register Set. 0 : none 1: Set	1b0
[5]	R/W	DETECTMODESET3_5	Alive GPIO5 High Level detect mode Register Set. 0 : none 1: Set	1b0
[4]	R/W	DETECTMODESET3_4	Alive GPIO4 High Level detect mode Register Set. 0 : none 1: Set	1b0
[3]	R/W	DETECTMODESET3_3	Alive GPIO3 High Level detect mode Register Set. 0 : none 1: Set	1b0
[2]	R/W	DETECTMODESET3_2	Alive GPIO2 High Level detect mode Register Set. 0 : none 1: Set	1b0
[1]	R/W	DETECTMODESET3_1	Alive GPIO1 High Level detect mode Register Set. 0 : none 1: Set	1b0
[0]	R/W	DETECTMODESET3_0	Alive GPIO0 High Level detect mode Register Set. 0 : none 1: Set	1b0

ALIVE GPIO High Level Detect Mode Read Register (ALIVEGPIOHIGHDETECTMODEREADREG)

Address : C001_0848h

[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOHIGHDETECT MODE7	Alive GPIO7 High Level detect mode register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIOHIGHDETECT MODE6	Alive GPIO6 High Level detect mode register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIOHIGHDETECT MODE5	Alive GPIO5 High Level detect mode register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIOHIGHDETECT MODE4	Alive GPIO4 High Level Async detect mode register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIOHIGHDETECT MODE3	Alive GPIO3 High Level detect mode register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIOHIGHDETECT MODE2	Alive GPIO2 High Level detect mode register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIOHIGHDETECT MODE1	Alive GPIO1 High Level detect mode register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIOHIGHDETECT MODE0	Alive GPIO0 High Level detect mode register 0 : Disable 1 : Enable	1b0

<Note> ALIVEGPIOHIGHDETECTMODE(n) Register operates as follows

It remains as the former state in case of {DETECTMODERST3_(n), DETECTMODESET3_(n)} = 2'b00

It is set as '1' in case of {DETECTMODERST3_(n), DETECTMODESET3_(n)} = 2'b01

It is set as '0' in case of {DETECTMODERST3_(n), DETECTMODESET3_(n)} = 2'b10

It is set as '1' in case of {DETECTMODERST3_(n), DETECTMODESET3_(n)} = 2'b11

Alive GPIO Detect Enable Reset Register (ALIVEPIODETECTENBRSTREG)

Address : C001_084Ch

[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTENBRST7	Alive GPIO7 Detect EnableRegister Reset. 0 : none 1: Reset	1b0

Bit	R/W	Symbol	Description	Reset Value
[6]	R/W	DETECTENBRST6	Alive GPIO6 Detect Enable Register Reset. 0 : none 1 : Reset	1b0
[5]	R/W	DETECTENBRST5	Alive GPIO5 Detect Enable Register Reset. 0 : none 1 : Reset	1b0
[4]	R/W	DETECTENBRST4	Alive GPIO4 Detect Enable Register Reset. 0 : none 1 : Reset	1b0
[3]	R/W	DETECTENBRST3	Alive GPIO3 Detect Enable Register Reset. 0 : none 1 : Reset	1b0
[2]	R/W	DETECTENBRST2	Alive GPIO2 Detect Enable Register Reset. 0 : none 1 : Reset	1b0
[1]	R/W	DETECTENBRST1	Alive GPIO1 Detect Enable Register Reset. 0 : none 1 : Reset	1b0
[0]	R/W	DETECTENBRST0	Alive GPIO0 Detect Enable Register Reset. 0 : none 1 : Reset	1b0

Allive GPIO Detect Enable Set Register (ALIVEGPIODETECTENBSETREG)

Address : C001_0850h

[31:8]	-	RESERVED	Reserved	0
[7]	R/W	DETECTENBSET7	Alive GPIO7 Detect Enable Register Set. 0 : none 1 : Set	1b0
[6]	R/W	DETECTENBSET6	Alive GPIO6 Detect Enable Register Set. 0 : none 1 : Set	1b0
[5]	R/W	DETECTENBSET5	Alive GPIO5 Detect Enable Register Set. 0 : none 1 : Set	1b0
[4]	R/W	DETECTENBSET4	Alive GPIO4 Detect Enable Register Set. 0 : none 1 : Set	1b0
[3]	R/W	DETECTENBSET3	Alive GPIO3 Detect Enable Register Set. 0 : none 1 : Set	1b0
[2]	R/W	DETECTENBSET2	Alive GPIO2 Detect Enable Register Set. 0 : none 1 : Set	1b0
[1]	R/W	DETECTENBSET1	Alive GPIO1 Detect Enable Register Set. 0 : none 1 : Set	1b0
[0]	R/W	DETECTENBSET0	Alive GPIO0 Detect Enable Register Set. 0 : none 1 : Set	1b0

ALIVE GPIO Detect Enable Read Register (ALIVEGPIODETECTENBREADREG)

Address : C001_0854h

[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIODETECTENB7	Alive GPIO7 Detect Enable register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIODETECTENB6	Alive GPIO6 Detect Enable register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIODETECTENB5	Alive GPIO5 Detect Enable register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIODETECTENB4	Alive GPIO4 Detect Enable register	1b0

Bit	R/W	Symbol	Description	Reset Value
			0 : Disable 1 : Enable	
[3]	R	ALIVEGPIODETECTENB3	Alive GPIO3 Detect Enable register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIODETECTENB2	Alive GPIO2 Detect Enable register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIODETECTENB1	Alive GPIO1 Detect Enable register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIODETECTENB0	Alive GPIO0 Detect Enable register 0 : Disable 1 : Enable	1b0

<Note> ALIVEGPIODETECTENB (n) Register operates as follows

It remains as the former state in case of {DETECTENBRST(n), DETECTENBSET(n)} = 2'b00

It is set as '1' in case of {DETECTENBRST(n), DETECTENBSET(n)} = 2'b01

It is set as '0' in case of {DETECTENBRST(n), DETECTENBSET(n)} = 2'b10

It is set as '1' in case of {DETECTENBRST(n), DETECTENBSET(n)} = 2'b11

Allive GPIO Interrupt Enable Reset Register (ALIVEGPIOINTENBRSTREG)

Address : C001_0858h

[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOINTENBRST7	Alive GPIO7 Interrupt EnableRegister Reset. 0 : none 1: Reset	1b0
[6]	R/W	ALIVEGPIOINTENBRST6	Alive GPIO6 Interrupt Enable Register Reset. 0 : none 1: Reset	1b0
[5]	R/W	ALIVEGPIOINTENBRST5	Alive GPIO5 Interrupt Enable Register Reset. 0 : none 1: Reset	1b0
[4]	R/W	ALIVEGPIOINTENBRST4	Alive GPIO4 Interrupt Enable Register Reset. 0 : none 1: Reset	1b0
[3]	R/W	ALIVEGPIOINTENBRST3	Alive GPIO3 Interrupt Enable Register Reset. 0 : none 1: Reset	1b0
[2]	R/W	ALIVEGPIOINTENBRST2	Alive GPIO2 Interrupt Enable Register Reset. 0 : none 1: Reset	1b0
[1]	R/W	ALIVEGPIOINTENBRST1	Alive GPIO1 Interrupt Enable Register Reset. 0 : none 1: Reset	1b0
[0]	R/W	ALIVEGPIOINTENBRST0	Alive GPIO0 Interrupt Enable Register Reset. 0 : none 1: Reset	1b0

Allive GPIO Detect Enable Set Register (ALIVEGPIODETECTENABLESETREG)

Address : C001_085Ch

[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOINTENBSET7	Alive GPIO7 Interrupt Enable Register Set. 0 : none 1: Set	1b0
[6]	R/W	ALIVEGPIOINTENBSET6	Alive GPIO6 Interrupt Enable Register Set. 0 : none 1: Set	1b0
[5]	R/W	ALIVEGPIOINTENBSET5	Alive GPIO5 Interrupt Enable Register Set. 0 : none 1: Set	1b0
[4]	R/W	ALIVEGPIOINTENBSET4	Alive GPIO4 Interrupt Enable Register Set.	1b0

Bit	R/W	Symbol	Description	Reset Value
			0 : none 1: Set	
[3]	R/W	ALIVEGPIOINTENBSET3	Alive GPIO3 Interrupt Enable Register Set. 0 : none 1: Set	1b0
[2]	R/W	ALIVEGPIOINTENBSET2	Alive GPIO2 Interrupt Enable Register Set. 0 : none 1: Set	1b0
[1]	R/W	ALIVEGPIOINTENBSET1	Alive GPIO1 Interrupt Enable Register Set. 0 : none 1: Set	1b0
[0]	R/W	ALIVEGPIOINTENBSET0	Alive GPIO0 Interrupt Enable Register Set. 0 : none 1: Set	1b0

ALIVE GPIO Interrut Enable Read Register (ALIVEGPIOINTENBREADREG)**Address : C001_0860h**

[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOINTENB7	Alive GPIO7 Interrupt Enable register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIOINTENB6	Alive GPIO6 Interrupt Enable register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIOINTENB5	Alive GPIO5 Interrupt Enable register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIOINTENB4	Alive GPIO4 Interrupt Enable register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIOINTENB3	Alive GPIO3 Interrupt Enable register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIOINTENB2	Alive GPIO2 Interrupt Enable register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIOINTENB1	Alive GPIO1 Interrupt Enable register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIOINTENB0	Alive GPIO0 Interrupt Enable register 0 : Disable 1 : Enable	1b0

<Note> ALIVEGPIOINTENB (n) Register operates as follows

It remains as the former state in case of {ALIVEGPIOENBRST(n), ALIVEGPIOENBSET(n)} = 2'b00

It is set as '1' in case of { ALIVEGPIOENBRST(n), ALIVEGPIOENBSET (n)} = 2'b01

It is set as '0' in case of { ALIVEGPIOENBRST(n), ALIVEGPIOENBSET (n)} = 2'b10

It is set as '1' in case of { ALIVEGPIOENBRST(n), ALIVEGPIOENBSET (n)} = 2'b11

ALIVE GPIO Detect Pending Register (ALIVEPIODETECTPENDREG)**Address : C001_0864h**

[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEPIODETECTPEN D7	Alive GPIO7 Detect Pending Read> 0:none 1 : Interrupt Pending Write> 0:none 1: Clear	1b0
[6]	R	ALIVEPIODETECTPEN D6	Alive GPIO6 Detect Pending Read> 0:none 1 : Interrupt Pending Write> 0:none 1: Clear	1b0
[5]	R	ALIVEPIODETECTPEN	Alive GPIO5 Detect Pending	1b0

Bit	R/W	Symbol	Description	Reset Value
		D5	Read> 0: none 1 : Interrupt Pending Write> 0: none 1: Clear	
[4]	R	ALIVEGPIODETECTPEN D4	Alive GPIO4 Detect Pending Read> 0: none 1 : Interrupt Pending Write> 0: none 1: Clear	1b0
[3]	R	ALIVEGPIODETECTPEN D3	Alive GPIO3 Detect Pending Read> 0: none 1 : Interrupt Pending Write> 0: none 1: Clear	1b0
[2]	R	ALIVEGPIODETECTPEN D2	Alive GPIO2 Detect Pending Read> 0: none 1 : Interrupt Pending Write> 0: none 1: Clear	1b0
[1]	R	ALIVEGPIODETECTPEN D1	Alive GPIO1 Detect Pending Read> 0: none 1 : Interrupt Pending Write> 0: none 1: Clear	1b0
[0]	R	ALIVEGPIODETECTPEN D0	Alive GPIO0 Detect Pending Read> 0: none 1 : Interrupt Pending Write> 0: none 1: Clear	1b0
<hr/>				
ALIVE SCRATCH RESET REGISTER (ALIVESCRATCHRSTREG)				
<i>Address : C001_0868h</i>				
[31 : 0]	R/W	ALIVESCRATCHRST	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST drives Scratch Register's reset pin.	32'b0
<hr/>				
ALIVE SCRATCH SET REGISTER (ALIVESCRATCHSETREG)				
<i>Address : C001_086Ch</i>				
[31 : 0]	R/W	ALIVESCRATCHSET	Set Alive Scratch Register Each bit of ALIVESCRATCHSET drives Scratch Register's set pin	32'b0
<hr/>				
ALIVE SCRATCH READ REGISTER (ALIVESCRATCHREADREG)				
<i>Address : C001_0870h</i>				
[31 : 0]	R	ALIVESCRATCHREAD	Read Alive Scratch Register	32'b0
<hr/>				
Alive GPIO PAD Out Enable Reset Register (ALIVEGPIOPADOUTENBRSTREG)				
<i>Address : C001_0874h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOPADOUTENB RST7	Alive GPIO7 PAD Out Enable Register Reset. 0 : none 1: Reset	1b0
[6]	R/W	ALIVEGPIOPADOUTENB RST6	Alive GPIO6 PAD Out Enable Register Reset. 0 : none 1: Reset	1b0
[5]	R/W	ALIVEGPIOPADOUTENB RST5	Alive GPIO5 PAD Out Enable Register Reset. 0 : none 1: Reset	1b0
[4]	R/W	ALIVEGPIOPADOUTENB RST4	Alive GPIO4 PAD Out Enable Register Reset. 0 : none 1: Reset	1b0
[3]	R/W	ALIVEGPIOPADOUTENB RST3	Alive GPIO3 PAD Out Enable Register Reset. 0 : none 1: Reset	1b0
[2]	R/W	ALIVEGPIOPADOUTENB	Alive GPIO2 PAD Out Enable Register Reset.	1b0

Bit	R/W	Symbol	Description	Reset Value
		RST2	0 : none 1: Reset	
[1]	R/W	ALIVEGPIOPADOUTENB RST1	Alive GPIO1 PAD Out Enable Register Reset. 0 : none 1: Reset	1b0
[0]	R/W	ALIVEGPIOPADOUTENB RST0	Alive GPIO0 PAD Out Enable Register Reset. 0 : none 1: Reset	1b0
Alive GPIO PAD Out Enable Set Register (ALIVEGPIOPADOUTENBSETREG)				
<i>Address : C001_0878h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOPADOUTENB SET7	Alive GPIO7 PAD Out Enable Register Set. 0 : none 1: Set	1b0
[6]	R/W	ALIVEGPIOPADOUTENB SET6	Alive GPIO6 PAD Out Enable Register Set. 0 : none 1: Set	1b0
[5]	R/W	ALIVEGPIOPADOUTENB SET5	Alive GPIO5 PAD Out Enable Register Set. 0 : none 1: Set	1b0
[4]	R/W	ALIVEGPIOPADOUTENB SET4	Alive GPIO4 PAD Out Enable Register Set. 0 : none 1: Set	1b0
[3]	R/W	ALIVEGPIOPADOUTENB SET3	Alive GPIO3 PAD Out Enable Register Set. 0 : none 1: Set	1b0
[2]	R/W	ALIVEGPIOPADOUTENB SET2	Alive GPIO2 PAD Out Enable Register Set. 0 : none 1: Set	1b0
[1]	R/W	ALIVEGPIOPADOUTENB SET1	Alive GPIO1 PAD Out Enable Register Set. 0 : none 1: Set	1b0
[0]	R/W	ALIVEGPIOPADOUTENB SET0	Alive GPIO0 PAD Out Enable Register Set. 0 : none 1: Set	1b0
ALIVE GPIO PAD Out Enable Read Register (ALIVEGPIOPADOUTENBREADREG)				
<i>Address : C001_087Ch</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOPADOUTENB 7	Alive GPIO7 PAD Out Enable register 0 : Disable 1 : Enable	1b0
[6]	R	ALIVEGPIOPADOUTENB 6	Alive GPIO6 PAD Out Enable register 0 : Disable 1 : Enable	1b0
[5]	R	ALIVEGPIOPADOUTENB 5	Alive GPIO5 PAD Out Enable register 0 : Disable 1 : Enable	1b0
[4]	R	ALIVEGPIOPADOUTENB 4	Alive GPIO4 PAD Out Enable register 0 : Disable 1 : Enable	1b0
[3]	R	ALIVEGPIOPADOUTENB 3	Alive GPIO3 PAD Out Enable register 0 : Disable 1 : Enable	1b0
[2]	R	ALIVEGPIOPADOUTENB 2	Alive GPIO2 PAD Out Enable register 0 : Disable 1 : Enable	1b0
[1]	R	ALIVEGPIOPADOUTENB 1	Alive GPIO1 PAD Out Enable register 0 : Disable 1 : Enable	1b0
[0]	R	ALIVEGPIOPADOUTENB 0	Alive GPIO0 PAD Out Enable register 0 : Disable 1 : Enable	1b0

Bit	R/W	Symbol	Description	Reset Value
<Note> ALIVEGPIOPADOUTENB(n) Register operates as follows				
It remains as the former state in case of {ALIVEGPIOPADOUTENBRST(n), ALIVEGPIOPADOUTENBSET (n)} = 2b00				
It is set as '1' in case of { ALIVEGPIOPADOUTENBRST (n), ALIVEGPIOPADOUTENBSET (n)} = 2b01				
It is set as '0' in case of { ALIVEGPIOPADOUTENBRST (n), ALIVEGPIOPADOUTENBSET (n)} = 2b10				
It is set as '1' in case of { ALIVEGPIOPADOUTENBRST (n), ALIVEGPIOPADOUTENBSET (n)} = 2b11				
Allive GPIO PAD Pullup Reset Register (ALIVEGPIOPADPULLUPRSTREG)				
Address : C001_0880h				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOPADPULLUP RST7	Alive GPIO7 PAD Pullup Register Reset. 0: none 1: Reset	1b0
[6]	R/W	ALIVEGPIOPADPULLUP RST6	Alive GPIO6 PAD Pullup Register Reset. 0: none 1: Reset	1b0
[5]	R/W	ALIVEGPIOPADPULLUP RST5	Alive GPIO5 PAD Pullup Register Reset. 0: none 1: Reset	1b0
[4]	R/W	ALIVEGPIOPADPULLUP RST4	Alive GPIO4 PAD Pullup Register Reset. 0: none 1: Reset	1b0
[3]	R/W	ALIVEGPIOPADPULLUP RST3	Alive GPIO3 PAD Pullup Register Reset. 0: none 1: Reset	1b0
[2]	R/W	ALIVEGPIOPADPULLUP RST2	Alive GPIO2 PAD Pullup Register Reset. 0: none 1: Reset	1b0
[1]	R/W	ALIVEGPIOPADPULLUP RST1	Alive GPIO1 PAD Pullup Register Reset. 0: none 1: Reset	1b0
[0]	R/W	ALIVEGPIOPADPULLUP RST0	Alive GPIO0 PAD Pullup Register Reset. 0: none 1: Reset	1b0
Allive GPIO PAD Pullup Set Register (ALIVEGPIOPADPULLUPSETREG)				
Address : C001_0884h				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOPADPULLUP SET7	Alive GPIO7 PAD Pullup Register Set. 0: none 1: Set	1b0
[6]	R/W	ALIVEGPIOPADPULLUP SET6	Alive GPIO6 PAD Pullup Register Set. 0: none 1: Set	1b0
[5]	R/W	ALIVEGPIOPADPULLUP SET5	Alive GPIO5 PAD Pullup Register Set. 0: none 1: Set	1b0
[4]	R/W	ALIVEGPIOPADPULLUP SET4	Alive GPIO4 PAD Pullup Register Set. 0: none 1: Set	1b0
[3]	R/W	ALIVEGPIOPADPULLUP SET3	Alive GPIO3 PAD Pullup Register Set. 0: none 1: Set	1b0
[2]	R/W	ALIVEGPIOPADPULLUP SET2	Alive GPIO2 PAD Pullup Register Set. 0: none 1: Set	1b0
[1]	R/W	ALIVEGPIOPADPULLUP SET1	Alive GPIO1 PAD Pullup Register Set. 0: none 1: Set	1b0
[0]	R/W	ALIVEGPIOPADPULLUP SET0	Alive GPIO0 PAD Pullup Register Set. 0: none 1: Set	1b0

Bit	R/W	Symbol	Description	Reset Value
ALIVE GPIO PAD Pullup Read Register (ALIVEGPIOPADPULLUPREADREG)				
<i>Address : C001_0888h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOPADPULLUP7	Alive GPIO7 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
[6]	R	ALIVEGPIOPADPULLUP6	Alive GPIO6 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
[5]	R	ALIVEGPIOPADPULLUP5	Alive GPIO5 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
[4]	R	ALIVEGPIOPADPULLUP4	Alive GPIO4 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
[3]	R	ALIVEGPIOPADPULLUP3	Alive GPIO3 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
[2]	R	ALIVEGPIOPADPULLUP2	Alive GPIO2 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
[1]	R	ALIVEGPIOPADPULLUP1	Alive GPIO1 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
[0]	R	ALIVEGPIOPADPULLUP0	Alive GPIO0 PAD Pullup register 0 : Pull-down 1 : Pull-up	1b1
<Note> ALIVEGPIOPADPULLUP (n) Register Register operates as follows				
It remains as the former state in case of {ALIVEGPIOPADPULLUPRST(n), ALIVEGPIOPADPULLUPSET (n)} = 2'b00				
It is set as '1' in case of { ALIVEGPIOPADOPULLUPRST (n), ALIVEGPIOPADPULLUPSET (n)} = 2'b01				
It is set as '0' in case of { ALIVEGPIOPADOPULLUPRST (n), ALIVEGPIOPADPULLUPSET (n)} = 2'b10				
It is set as '1' in case of { ALIVEGPIOPADOPULLUPRST (n), ALIVEGPIOPADPULLUPSET (n)} = 2'b11				
Alive GPIO PAD Out Reset Register (ALIVEGPIOPADOUTRSTREG)				
<i>Address : C001_088Ch</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOPADOUTRST7	Alive GPIO7 PAD Out Register Reset. 0 : none 1: Reset	1b0
[6]	R/W	ALIVEGPIOPADOUTRST6	Alive GPIO6 PAD Out Register Reset. 0 : none 1: Reset	1b0
[5]	R/W	ALIVEGPIOPADOUTRST5	Alive GPIO5 PAD Out Register Reset. 0 : none 1: Reset	1b0
[4]	R/W	ALIVEGPIOPADOUTRST4	Alive GPIO4 PAD Out Register Reset. 0 : none 1: Reset	1b0
[3]	R/W	ALIVEGPIOPADOUTRST3	Alive GPIO3 PAD Out Register Reset. 0 : none 1: Reset	1b0
[2]	R/W	ALIVEGPIOPADOUTRST2	Alive GPIO2 PAD Out Register Reset. 0 : none 1: Reset	1b0
[1]	R/W	ALIVEGPIOPADOUTRST1	Alive GPIO1 PAD Out Register Reset. 0 : none 1: Reset	1b0
[0]	R/W	ALIVEGPIOPADOUTRST0	Alive GPIO0 PAD Out Register Reset. 0 : none 1: Reset	1b0

Bit	R/W	Symbol	Description	Reset Value
Allive GPIO PAD Out Set Register (ALIVEGPIOPADOUTSETREG)				
<i>Address : C001_0890h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R/W	ALIVEGPIOPADOUTSET7	Alive GPIO7 PAD Out Register Set. 0 : none 1: Set	1b0
[6]	R/W	ALIVEGPIOPADOUTSET6	Alive GPIO6 PAD Out Register Set. 0 : none 1: Set	1b0
[5]	R/W	ALIVEGPIOPADOUTSET5	Alive GPIO5 PAD Out Register Set. 0 : none 1: Set	1b0
[4]	R/W	ALIVEGPIOPADOUTSET4	Alive GPIO4 PAD Out Register Set. 0 : none 1: Set	1b0
[3]	R/W	ALIVEGPIOPADOUTSET3	Alive GPIO3 PAD Out Register Set. 0 : none 1: Set	1b0
[2]	R/W	ALIVEGPIOPADOUTSET2	Alive GPIO2 PAD Out Register Set. 0 : none 1: Set	1b0
[1]	R/W	ALIVEGPIOPADOUTSET1	Alive GPIO1 PAD Out Register Set. 0 : none 1: Set	1b0
[0]	R/W	ALIVEGPIOPADOUTSET0	Alive GPIO0 PAD Out Register Set. 0 : none 1: Set	1b0
ALIVE GPIO PAD Out Read Register (ALIVEGPIOPADOUTREADREG)				
<i>Address : C001_0894h</i>				
[31:8]	-	RESERVED	Reserved	0
[7]	R	ALIVEGPIOPADOUT7	Alive GPIO7 PAD Out register	1b0
[6]	R	ALIVEGPIOPADOUT6	Alive GPIO6 PAD Out register	1b0
[5]	R	ALIVEGPIOPADOUT5	Alive GPIO5 PAD Out register	1b0
[4]	R	ALIVEGPIOPADOUT4	Alive GPIO4 PAD Out register	1b0
[3]	R	ALIVEGPIOPADOUT3	Alive GPIO3 PAD Out register	1b0
[2]	R	ALIVEGPIOPADOUT2	Alive GPIO2 PAD Out register	1b0
[1]	R	ALIVEGPIOPADOUT1	Alive GPIO1 PAD Out register	1b0
[0]	R	ALIVEGPIOPADOUT0	Alive GPIO0 PAD Out register	1b0
<Note> ALIVEGPIOPADOUT(n) Register operates as follows				
It remains as the former state in case of {ALIVEGPIOPADOUTRST(n), ALIVEGPIOPADOUTSET (n)} = 2'b00				
It is set as '1' in case of { ALIVEGPIOPADOUTRST (n), ALIVEGPIOPADOUTSET (n)} = 2'b01				
It is set as '0' in case of { ALIVEGPIOPADOUTRST (n), ALIVEGPIOPADOUTSET (n)} = 2'b10				
It is set as '1' in case of { ALIVEGPIOPADOUTRST (n), ALIVEGPIOPADOUTSET (n)} = 2'b11				
VDD Control reset Register (VDDCTRLRSTREG)				
<i>Address : C001_0898h</i>				
[31:10]	-	RESERVED	Reserved	0
[9]	R/W	PADHOLDENBRST3	nPADHOLDENB3 Reset This register is not used in current version. Write to this register does not affect anything. 0 : none 1: Set	1b0

Bit	R/W	Symbol	Description	Reset Value
[8]	R/W	PADHOLDENBRST2	nPADHOLDENB2 Reset 0: none 1: Set	1b0
[7]	R/W	PADHOLDENBRST1	nPADHOLDENB1 Reset 0: none 1: Set	1b0
[6]	R/W	PADHOLDENBRST0	nPADHOLDENB0 Reset 0: none 1: Set	1b0
[5]	R/W	PADHOLDRST3	nPADHOLD3 Reset This register is not used in current version. Write to this register does not affect anything. 0: none 1: Set	1b0
[4]	R/W	PADHOLDRST2	nPADHOLD2 Reset 0: none 1: Set	1b0
[3]	R/W	PADHOLDRST1	nPADHOLD1 Reset 0: none 1: Set	1b0
[2]	R/W	PADHOLDRST0	nPADHOLD0 Reset 0: none 1: Set	1b0
[1]	R/W	VDDPWRONRST_DDR	DRAM VDD Power ON Reset 0: none 1: Set	1b0
[0]	R/W	VDDPWRONRST	Core VDD Power ON Reset 0: none 1: Set	1b0

VDD Control set Register (VDDCTRLSETREG)

Address : C001_089Ch

[31:9]	-	RESERVED	Reserved	0
[9]	R/W	PADHOLDENBSET3	nPADHOLDENB3 Set This register is not used in current version. Write to this register does not affect anything. 0: none 1: Set	1b0
[8]	R/W	PADHOLDENBSET2	nPADHOLDENB2 Set 0: none 1: Set	1b0
[7]	R/W	PADHOLDENBSET1	nPADHOLDENB1 Set 0: none 1: Set	1b0
[6]	R/W	PADHOLDENBSET0	nPADHOLDENB0 Set 0: none 1: Set	1b0
[5]	R/W	PADHOLDSET3	nPADHOLD3 Set This register is not used in current version. Write to this register does not affect anything. 0: none 1: Set	1b0
[4]	R/W	PADHOLDSET2	nPADHOLD2 Set 0: none 1: Set	1b0
[3]	R/W	PADHOLDSET1	nPADHOLD1 Set 0: none 1: Set	1b0
[2]	R/W	PADHOLDSET0	nPADHOLD0 Set 0: none 1: Set	1b0
[1]	R/W	VDDPWRONSET_DDR	DRAM VDD Power ON Set 0: none 1: Set	1b0
[0]	R/W	VDDPWRONSET	Core VDD Power ON Set 0: none 1: Set	1b0

Bit	R/W	Symbol	Description	Reset Value
VDD Control Read Register (VDDCTRLREADREG)				
<i>Address : C001_08A0h</i>				
[31:11]	-	RESERVED	Reserved	0
[10]	R	VDDPWRTOGGLE	Read VDDPWRTOGGLE PAD status 0: user does not pushed VDDPWRTOGGLE PAD, 1: user pushed VDDPWRTOGGLE PAD	-
[9]	R	NPADHOLDENB3	Read nPADHOLDENB3 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[8]	R	NPADHOLDENB2	Read nPADHOLDENB2 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[7]	R	NPADHOLDENB1	Read nPADHOLDENB1 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[6]	R	NPADHOLDENB0	Read nPADHOLDENB0 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[5]	R	NPADHOLD2	Read nPADHOLD2 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[4]	R	NPADHOLD2	Read nPADHOLD2 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[3]	R	NPADHOLD1	Read nPADHOLD1 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[2]	R	NPADHOLD0	Read nPADHOLD0 Register 0: Pad Retention Enable 1: Pad Retention Disable	1b1
[1]	R	VDDPWRON_DDR	Read DRAM Vdd Power On Register 0: DRAM Power Off 1: CORE DRAM On	1b1
[0]	R	VDDPWRON	Read CORE Vdd Power On Register 0: CORE Power Off 1: CORE Power On	1b1
<Note0> Each bit of VDDCTRLREADREG is set or reset by each bit of VDDCTRLSETREG/VDDCTRLRSTREG				
<Note1> The Pad Retention of nPadHod and nPadHoldEnb operate as follows.				
Pad Retention Disable if nPadHod == '1', nPadHoldEnb=='1' when Core Power turns off				
Pad Retention Disable if nPadHod == '0', nPadHoldEnb=='1' when Core Power turns off				
Pad Retention Enable if nPadHod == '1', nPadHoldEnb=='0' when Core Power turns off				
Pad Retention Enable if nPadHod == '0', nPadHoldEnb=='0' when Core Power turns off				
Pad Retention Disable if nPadHod == '1', nPadHoldEnb=='1' when Core Power turns on				
Pad Retention Disable if nPadHod == '0', nPadHoldEnb=='1' when Core Power turns on				
Pad Retention Disable if nPadHod == '1', nPadHoldEnb=='0' when Core Power turns on				
Pad Retention Enable if nPadHod == '0', nPadHoldEnb=='0' when Core Power turns on				
ALIVE CLEAR WAKEUP STATUS REGISTER (ALIVECLEARWAKEUPSTATUSREGISTER)				
<i>Address : C001_08A4h</i>				
[31:1]	R	RESERVED	reserved	31h0
[0]	W	CLRWAKEUP	Clear is wakeup status register 0: none 1: clear	1b0
ALIVE SLEEP WAKEUP STATUS REGISTER (ALIVESLEEPWAKEUPSTATUSREGISTER)				
<i>Address : C001_08A8h</i>				
[31:10]	R	RESERVED	reserved	22b0

Bit	R/W	Symbol	Description	Reset Value
[9]	R	ALIVEGPIO7	0 : none 1 : wakeup is ALIVEGPIO[7]	1'b0
[8]	R	ALIVEGPIO6	0 : none 1 : wakeup is ALIVEGPIO[6]	1'b0
[7]	R	ALIVEGPIO5	0 : none 1 : wakeup is ALIVEGPIO[5]	1'b0
[6]	R	ALIVEGPIO4	0 : none 1 : wakeup is ALIVEGPIO[4]	1'b0
[5]	R	ALIVEGPIO3	0 : none 1 : wakeup is ALIVEGPIO[3]	1'b0
[4]	R	ALIVEGPIO2	0 : none 1 : wakeup is ALIVEGPIO[2]	1'b0
[3]	R	ALIVEGPIO1	0 : none 1 : wakeup is ALIVEGPIO[1]	1'b0
[2]	R	ALIVEGPIO0	0 : none 1 : wakeup is ALIVEGPIO[0]	1'b0
[1]	R	RTCINTERRUPT	0 : none 1 : wakeup is rtc interrupt	1'b0
[0]	R	NVDDPWRTOGGLE	0 : none 1 : vddpwr toggle	1'b0
ALIVE SCRATCH RESET REGISTER1 (ALIVESCRATCHRSTREG1)				
<i>Address : C001_08ACh</i>				
[31 : 0]	R/W	ALIVESCRATCHRST1	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST1 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER1 (ALIVESCRATCHRSTREG1)				
<i>Address : C001_08B0h</i>				
[31 : 0]	R/W	ALIVESCRATCHSET1	Set Alive Scratch Register Each bit of ALIVESCRATCHSET1 drives Scratch Register's set pin.	32'b0
ALIVE SCRATCH READ REGISTER1 (ALIVESCRATCHRSTREG1)				
<i>Address : C001_08B4h</i>				
[31 : 0]	R	ALIVESCRATCHREAD1	Read Alive Scratch Register	32'b0
ALIVE SCRATCH RESET REGISTER2 (ALIVESCRATCHRSTREG2)				
<i>Address : C001_08B8h</i>				
[31 : 0]	R/W	ALIVESCRATCHRST2	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST2 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER2 (ALIVESCRATCHRSTREG2)				
<i>Address : C001_08BCh</i>				
[31 : 0]	R/W	ALIVESCRATCHSET2	Set Alive Scratch Register Each bit of ALIVESCRATCHSET2 drives Scratch Register's set pin.	32'b0
ALIVE SCRATCH READ REGISTER2 (ALIVESCRATCHRSTREG2)				
<i>Address : C001_08C0h</i>				
[31 : 0]	R	ALIVESCRATCHREAD2	Read Alive Scratch Register	32'b0
ALIVE SCRATCH RESET REGISTER3 (ALIVESCRATCHRSTREG3)				
<i>Address : C001_08C4h</i>				
[31 : 0]	R/W	ALIVESCRATCHRST3	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST3 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER3 (ALIVESCRATCHRSTREG3)				
<i>Address : C001_08C8h</i>				
[31 : 0]	R/W	ALIVESCRATCHSET3	Set Alive Scratch Register Each bit of ALIVESCRATCHSET3 drives Scratch Register's set pin.	32'b0
ALIVE SCRATCH READ REGISTER3 (ALIVESCRATCHRSTREG3)				

Bit	R/W	Symbol	Description	Reset Value
Address : C001_08CCh				
[31 : 0]	R	ALIVESCRATCHREAD3	Read Alive Scratch Register	32'b0
ALIVE SCRATCH RESET REGISTER4 (ALIVESCRATCHRSTREG4)				
Address : C001_08D0h				
[31 : 0]	R/W	ALIVESCRATCHRST4	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST4 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER4 (ALIVESCRATCHRSTREG4)				
Address : C001_08D4h				
[31 : 0]	R/W	ALIVESCRATCHSET4	Set Alive Scratch Register Each bit of ALIVESCRATCHSET4 drives Scratch Register's set pin.	32'b0
ALIVE SCRATCH READ REGISTER4 (ALIVESCRATCHRSTREG4)				
Address : C001_08D8h				
[31 : 0]	R	ALIVESCRATCHREAD4	Read Alive Scratch Register	32'b0
ALIVE SCRATCH RESET REGISTER5 (ALIVESCRATCHRSTREG5)				
Address : C001_08DCh				
[31 : 0]	R/W	ALIVESCRATCHRST5	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST5 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER5 (ALIVESCRATCHRSTREG5)				
Address : C001_08E0h				
[31 : 0]	R/W	ALIVESCRATCHSET5	Set Alive Scratch Register Each bit of ALIVESCRATCHSET5 drives Scratch Register's set pin.	32'b0
ALIVE SCRATCH READ REGISTER5 (ALIVESCRATCHRSTREG5)				
Address : C001_08E4h				
[31 : 0]	R	ALIVESCRATCHREAD5	Read Alive Scratch Register	32'b0
ALIVE SCRATCH RESET REGISTER6 (ALIVESCRATCHRSTREG6)				
Address : C001_08E8h				
[31 : 0]	R/W	ALIVESCRATCHRST5	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST6 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER6 (ALIVESCRATCHRSTREG6)				
Address : C001_08EcH				
[31 : 0]	R/W	ALIVESCRATCHSET6	Set Alive Scratch Register Each bit of ALIVESCRATCHSET6 drives Scratch Register's set pin.	32'b0
ALIVE SCRATCH READ REGISTER6 (ALIVESCRATCHRSTREG6)				
Address : C001_08F0h				
[31 : 0]	R	ALIVESCRATCHREAD6	Read Alive Scratch Register	32'b0
ALIVE SCRATCH RESET REGISTER7 (ALIVESCRATCHRSTREG7)				
Address : C001_08F4h				
[31 : 0]	R/W	ALIVESCRATCHRST7	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST7 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER5 (ALIVESCRATCHRSTREG7)				
Address : C001_08F8h				
[31 : 0]	R/W	ALIVESCRATCHSET7	Set Alive Scratch Register Each bit of ALIVESCRATCHSET7 drives Scratch Register's set pin.	32'b0

Bit	R/W	Symbol	Description	Reset Value
ALIVE SCRATCH READ REGISTER7 (ALIVESCRATCHRSTREG7)				
<i>Address : C001_08FCh</i>				
[31 : 0]	R	ALIVESCRATCHREAD7	Read Alive Scratch Register	32'b0
ALIVE SCRATCH RESET REGISTER8 (ALIVESCRATCHRSTREG8)				
<i>Address : C001_0900h</i>				
[31 : 0]	R/W	ALIVESCRATCHRST5	Reset Alive Scratch Register Each bit of ALIVESCRATCHRST3 drives Scratch Register's reset pin.	32'b0
ALIVE SCRATCH SET REGISTER8 (ALIVESCRATCHRSTREG8)				
<i>Address : C001_0904h</i>				
[31 : 0]	R/W	ALIVESCRATCHSET8	Set Alive Scratch Register Each bit of ALIVESCRATCHSET8 drives Scratch Register's set pin.	32'b0
ALIVE SCRATCH READ REGISTER8 (ALIVESCRATCHRSTREG8)				
<i>Address : C001_0908h</i>				
[31 : 0]	R	ALIVESCRATCHREAD8	Read Alive Scratch Register	32'b0
VDD OFF DELAY RESET REGISTER (VDOFFDELAYRSTREGISTER)				
<i>Address : C001_090Ch</i>				
[31 : 0]	R/W	VDOFFDELAYRST	Reset VDD Off delay value register Each bit of vddoffdelayrst drives Vdd off delay value Register's reset pin.	32'b0
VDD OFF DELAY SET REGISTER (VDOFFDELAYSETREGISTER))				
<i>Address : C001_0910h</i>				
[31 : 0]	R/W	VDOFFDELAYSET	Set VDD Off delay value register Each bit of vddoffdelayset drives Vdd off delay value Register's set pin.	32'b0
VDD OFF DELAY VALUE REGISTER (VDOFFDELAYVALUEREGISTER)				
<i>Address : C001_0914h</i>				
[31 : 0]	R	VDOFFDELAYVALUE	VDD OFF DELAY VALUE(unit : RTC clock)	32'b0
VDD OFF DELAY TIME REGISTER (VDOFFDELAYTIMEREREGISTER)				
<i>Address : C001_0918h</i>				
[31 : 0]	R	VDOFFDELAYTIME	VDD OFF DELAY TIME(unit : RTC clock) clear is vddoffdelayrst/set register write	32'b0
ALIVE GPIO INPUT VALUE READ REGISTER (ALIVEGPIOINPUTVALUE)				
<i>Address : C001_0918h</i>				
[31 : 0]	R	ALIVEGPIOINPUTVALUE	Nth bit of this register show the value of AliveGPIO[N].	32'b0
Reserved				
<i>Address : ~C001_0BFFh</i>				
RTC (See RTC section.)				
<i>Address : C001_0C00h ~ C001_0CFFh</i>				
PMU NISOLATE REGISTER (PMUNISOLATE)				
<i>Address : C001_0D00h</i>				
[31:2]	-	RESERVED	Reserved	0
[1]	R/W	NISOLATE_MFC	Power isolation (negative active Power Isolation) for MFC block The default value is 1 (all blocks are connected normally) Sub-block must be isolatioted in order to power off the sub-block 0 : Isolate (ready-to-power-off) 1 : normal mode	1

Bit	R/W	Symbol	Description	Reset Value
[0]	R/W	NISOLATE_GPU	Power isolation (negative active Power Isolation) for GPU block The default value is 1 (all blocks are connected normally) Sub-block must be isolated in order to power off the sub-blocks 0 : Isolate (ready-to-power-off) 1 : normal mode	1

NPOWERGATING should be set to '1' to access this register.

PMU POWER UP PRECHARGE REGISTER (PMUPWRUPPRE)

Address : C001_0D04h

[31:2]	-	RESERVED	Reserved	0
[1]	R/W	PWRUPPRE_MFC	Power up precharge for MFC block Sub-block must be precharged before powering up all power switches. 0 : Power down precharge 1 : Power up precharge	0
[0]	R/W	PWRUPPRE_GPU	Power up precharge for GPU block Sub-block must be precharged before powering up all power switches. 0 : Power down precharge 1 : Power up precharge	0

NPOWERGATING should be set to '1' to access this register.

PMU POWER UP REGISTER (PMUPWRUP)

Address : C001_0D08h

[31:2]	-	RESERVED	Reserved	0
[1]	R/W	PWRUP_MFC	Power up for MFC block Sub-blocks must be powered up all switches before normal operation. 0 : Power down 1 : Power up	0
[0]	R/W	PWRUP_GPU	Power up for GPU block Sub-blocks must be powered up all switches before normal operation. 0 : Power down 1 : Power up	0

NPOWERGATING should be set to '1' to access this register.

PMU POWER UP ACK REGISTER (PMUPWRUPACK)

Address : C001_0D0Ch

[31:2]	-	RESERVED	Reserved	0
[1]	R	PWRUPACK_MFC	Power up acknowledge for MFC block This register must be checked after after PMUPWRUP register is set to power up. 0 : Power Off 1 : Power On (Powering Up Sequence is finished)	0
[0]	R	PWRUPACK_GPU	Power up acknowledge for GPU block This register must be checked after after PMUPWRUP register is set to power up. 0 : Power Off 1 : Power On (Powering Up Sequence is finished)	0

Section 13. ID Register

13.1 Overview

ECID module of the NXP4330D/Q stores the 128bit DIEID information on a e-fuse ROM. Each Chip will have its own DIEID to identify it.

13.1.1 Features

- Support 128 bit Die ID
- Support 128 bit Secure Boot ID
- Support 128 bit Secure JTAG ID
- Support 128 bit Backdoor JTAG ID
- Programmable Secure Boot ID, Secure JTAG ID, Backdoor JTAG ID

13.2 Functional Description

13.2.1 AC Timing

Unit : ns

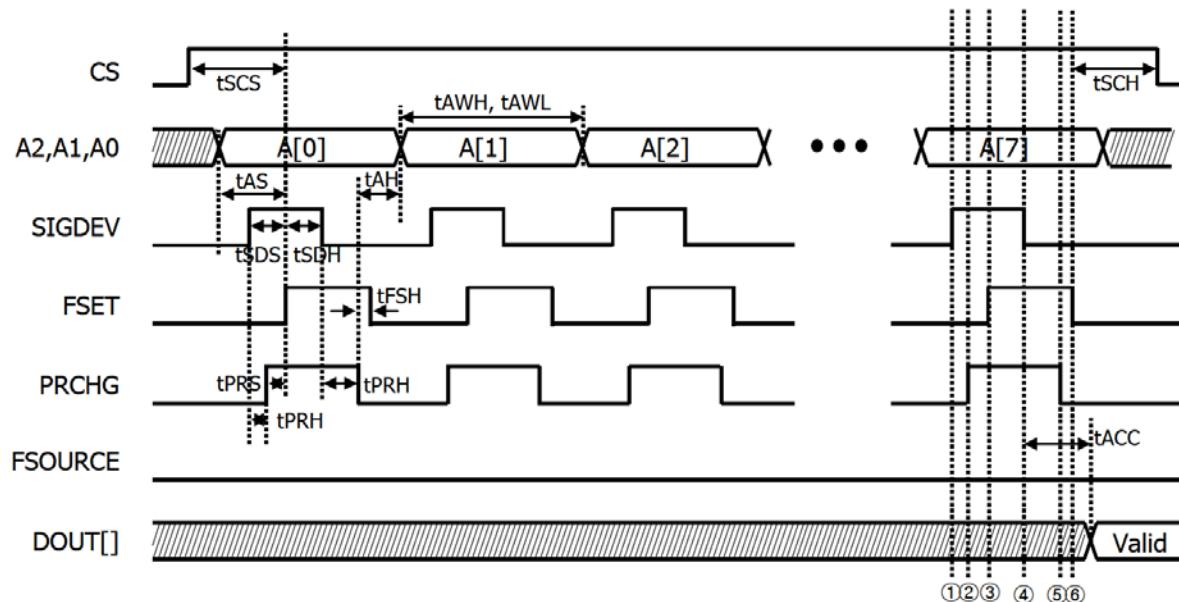
Symbol	Description	Symbol	Description
Program Mode			
tPCS	CS to PROG setup time (setup rising, rist constraint)	tPCH	CS to PROG hold time (hold falling, fall constraint)
tFSRCS	FSOURCE to PROG setup time (setup rising, rise constraint)	tFSRCH	FSOURCE to PROG hold time (hold falling, fall constraint)
tPRW	PROG pulse width high (rising edge)	tSCW	SCK pulse width high (rising edge)
tPSS	SCK to PROG setup time (setup rising, fall constraint)	tPSH	SCK to PROG hold time (hold falling, rise constraint)
tSIS	SDI to SCK setup time (setup falling, rise/fall constraint)	tSIH	SDI to SCK hold time (hold falling, rise/fall constraint)
tSAS	SDI to A0 setup time (setup rising, rise constraint)	tSAH	SDI to A0 hold time (hold falling, fall constraint)
tAAS	A2, A1 to A0 setup time (setup rising, rising constraint)	tAAH	A2, A1 to A0 hold time (hold falling, fall constraint)
tAWH	A0 pulse width high	tAWL	A0 pulse width low
tPAS	A0 to PROG setup time (setup rising, fall constraint)	tPAH	A0 to PROG hold time (hold falling, rise constraint)
Sense Mode			
tSCS	CS to FSET setup time (setup rising, rise constraint)	tSCH	CS to FSET hold time (hold falling, fall constraint)
tAS	ADDR[] to FSET setup time (setup rising, rise/fall constraint)	tAH	ADDR[] to PRCHG hold time (hold falling, rise/fall constraint)
tPRS	PRCHG to FSET setup time (setup rising, rise constraint)	tPRH	PRCHG to SIGDEV hold time (hold rising, rise constraint)
			PRCHG to SIGDEV hold time (hold falling, fall constraint)
tSDS	SIGDEV to FSET setup time (setup rising, rise constraint)	tSDH	SIGDEV to FSET hold time (hold rising, fall constraint)
tACC	DOUT[] access time after SIGDEV fall (falling edge)	tFSH	FSET to PRCHG hold time (hold falling, fall constraint)
Scan Mode			
tDCS	CS to SCK setup time (setup rising, rise constraint)	tDCH	CS to SCK hold time (hold falling, fall constraint)
tPACC	DOUT[] access time after SCK fall (falling edge)	tSACC	SDOUT access time after SCK rise (rising edge)

Table 13-1 AC Timing Table

- Blue : It will use each program mode and scan mode
- Red : It will use each program mode and sense mode
- tFSRCS, tFSRCH = 1000ns, All other timing arcs = 2ns
- tPRW = 10000ns ± 100ns (Not allowed out of tPRW range. Must need approval from PTE/DT team if any change)

\$\$\$ Figure 1-3 shows the characteristics of the vertical filter of Scaler and the filter has the setting range between 0 and 31. The Scaler register, SCCFGREG.SC_VFILT_COEFF, is used for the setting.

13.2.2 Sense Mode Timing



[Caution]

Keep the timing transition sequence during each address period: ①→②→③→④→⑤→⑥

Read fail at timing violation

- PROG, SCK = "Low"
- SDI = "Low"

Figure 13-1 Sense mode timing

- When SIGDEV is enabled ("H"), sensing current is occurred. So you should reduce SIGDEV pulse width high period for power saving.
- CS should be set high (enable) and A2, A1, A0 states should be set properly before first SIGDEV rising in sense mode.
- After sense operation, DOUT[] have "0" for un-blown fuse cell and "1" for blown fuse cell.
- Sense (read) operation would be performed correctly with timing diagram above. The user who want to have different timing sequence must have a prior consultation with effuse designer.

13.2.3 Scan-latch Mode Timing

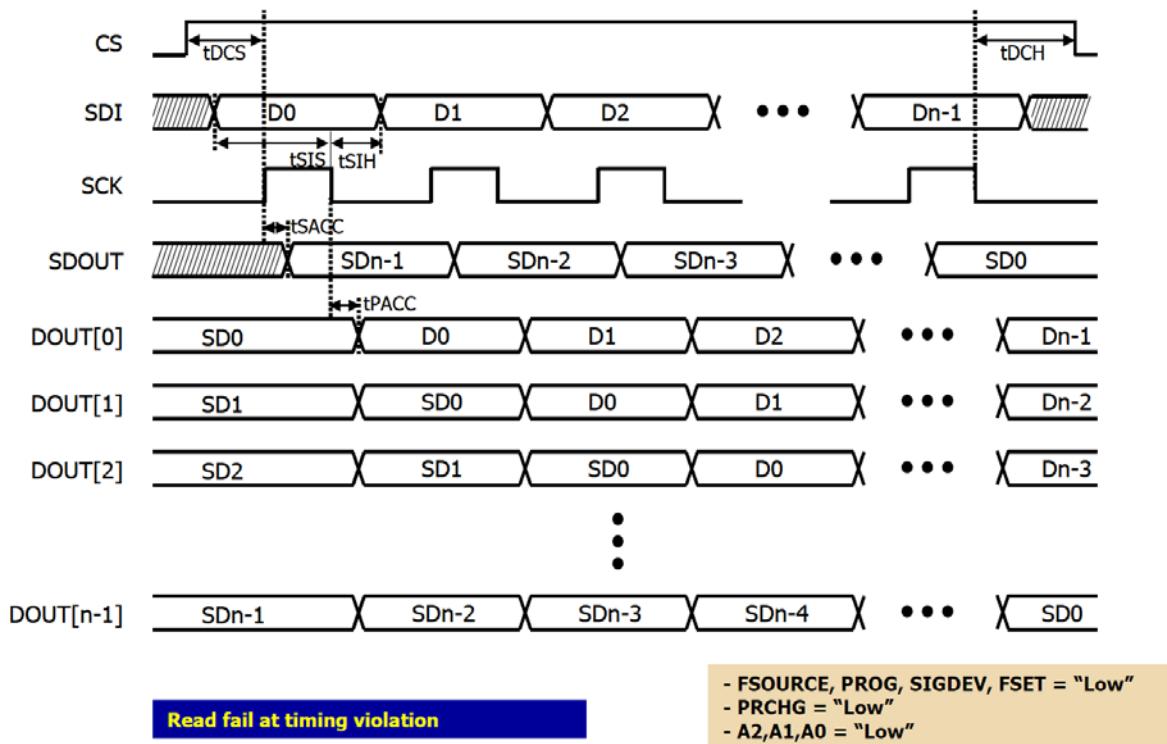


Figure 13-2 Scan-latch mode timing

- After sense mode, cell's data is shifted to SDOUT by SCK in Scan-latch mode
- Scan-latch operation would be performed correctly with timing diagram above. The user who want to have different timing sequence must have a prior consultation with effuse designer.

13.2.4 Stand-by Mode Timing

- CS pin should be set to ground (=0.0v) to reduce unnecessary leakage current.
- Other control pins : don't care.

13.2.5 Program Mode Timing

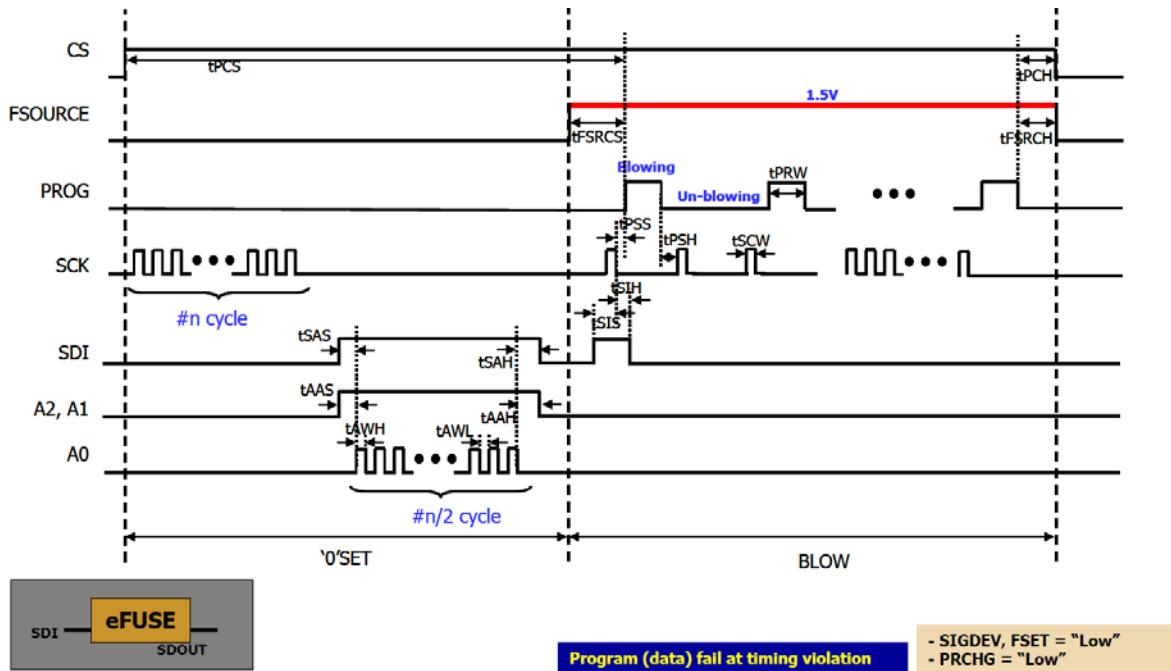


Figure 13-3 Fuse programming operation using single macro

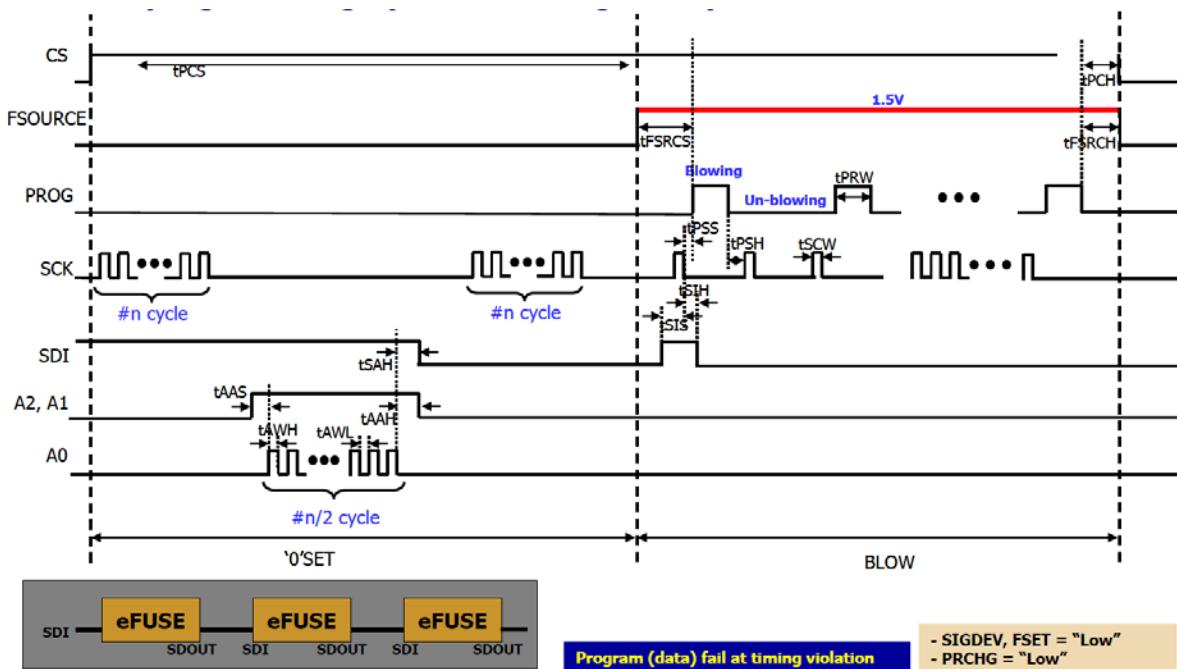


Figure 13-4 Fuse programming operation using single macro

- Apply $t_{FSRCS} > 1\mu s$ because of prevention leakage current from FSORUCE to ESD protection diode.
- Fuse program operation would be performed correctly with timing diagram above. The user who want to have different timing sequence must have a prior consultation with effuse designer.

13.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
128bit ECID Register 0 (ECID0)				
Address : C006_7000h				
[31:0]	R	ECID0	128bit ECID Register [31:0]	
128bit ECID Register 1 (ECID1)				
Address : C006_7004h				
[31:0]	R	ECID1	128bit ECID Register [63:32]	
128bit ECID Register 2 (ECID2)				
Address : C006_7008h				
[31:0]	R	ECID2	128bit ECID Register [95:64]	
128bit ECID Register 3 (ECID3)				
Address : C006_700Ch				
[31:0]	R	ECID3	128bit ECID Register [127:96]	
Chip Name Register 03_00(CHIP_NAME_03_00)				
Address : C006_7010h				
[31:24]	R	CHIP_NAME_3	Chip name character 3	
[23:16]	R	CHIP_NAME_2	Chip name character 2	
[15:8]	R	CHIP_NAME_1	Chip name character 1	
[7:0]	R	CHIP_NAME_0	Chip name character 0	
Chip Name Register 07_04(CHIP_NAME_07_04)				
Address : C006_7014h				
[31:24]	R	CHIP_NAME_7	Chip name character 7	
[23:16]	R	CHIP_NAME_6	Chip name character 6	
[15:8]	R	CHIP_NAME_5	Chip name character 5	
[7:0]	R	CHIP_NAME_4	Chip name character 4	
Chip Name Register 11_08(CHIP_NAME_11_08)				
Address : C006_7018h				
[31:24]	R	CHIP_NAME_11	Chip name character 11	
[23:16]	R	CHIP_NAME_10	Chip name character 10	
[15:8]	R	CHIP_NAME_9	Chip name character 9	
[7:0]	R	CHIP_NAME_8	Chip name character 8	
Chip Name Register 15_12(CHIP_NAME_15_12)				
Address : C006_701Ch				
[31:24]	R	CHIP_NAME_15	Chip name character 15	
[23:16]	R	CHIP_NAME_14	Chip name character 14	
[15:8]	R	CHIP_NAME_13	Chip name character 13	

Bit	R/W	Symbol	Description	Reset Value
[7:0]	R	CHIP_NAME_12	Chip name character 12	
Chip Name Register 19_16(CHIP_NAME_19_16)				
Address : C006_7020h				
[31:24]	R	CHIP_NAME_19	Chip name character 19	
[23:16]	R	CHIP_NAME_18	Chip name character 18	
[15:8]	R	CHIP_NAME_17	Chip name character 17	
[7:0]	R	CHIP_NAME_16	Chip name character 16	
Chip Name Register 23_20(CHIP_NAME_23_20)				
Address : C006_7024h				
[31:24]	R	CHIP_NAME_23	Chip name character 23	
[23:16]	R	CHIP_NAME_22	Chip name character 22	
[15:8]	R	CHIP_NAME_21	Chip name character 21	
[7:0]	R	CHIP_NAME_20	Chip name character 20	
Chip Name Register 27_24(CHIP_NAME_27_24)				
Address : C006_7028h				
[31:24]	R	CHIP_NAME_27	Chip name character 27	
[23:16]	R	CHIP_NAME_26	Chip name character 26	
[15:8]	R	CHIP_NAME_25	Chip name character 25	
[7:0]	R	CHIP_NAME_24	Chip name character 24	
Chip Name Register 31_28(CHIP_NAME_31_28)				
Address : C006_702Ch				
[31:24]	R	CHIP_NAME_31	Chip name character 31	
[23:16]	R	CHIP_NAME_30	Chip name character 30	
[15:8]	R	CHIP_NAME_29	Chip name character 29	
[7:0]	R	CHIP_NAME_28	Chip name character 28	
Chip Name Register 35_32(CHIP_NAME_35_32)				
Address : C006_7030h				
[31:24]	R	CHIP_NAME_35	Chip name character 35	
[23:16]	R	CHIP_NAME_34	Chip name character 34	
[15:8]	R	CHIP_NAME_33	Chip name character 33	
[7:0]	R	CHIP_NAME_32	Chip name character 32	
Chip Name Register 39_36(CHIP_NAME_39_36)				
Address : C006_7034h				
[31:24]	R	CHIP_NAME_39	Chip name character 39	
[23:16]	R	CHIP_NAME_38	Chip name character 38	
[15:8]	R	CHIP_NAME_37	Chip name character 37	
[7:0]	R	CHIP_NAME_36	Chip name character 36	

Bit	R/W	Symbol	Description	Reset Value
Chip Name Register 43_40(CHIP_NAME_43_40)				
Address : C006_7038h				
[31:24]	R	CHIP_NAME_43	Chip name character 43	
[23:16]	R	CHIP_NAME_42	Chip name character 42	
[15:8]	R	CHIP_NAME_41	Chip name character 41	
[7:0]	R	CHIP_NAME_40	Chip name character 40	
Chip Name Register 47_44(CHIP_NAME_47_44)				
Address : C006_703Ch				
[31:24]	R	CHIP_NAME_47	Chip name character 47	
[23:16]	R	CHIP_NAME_46	Chip name character 46	
[15:8]	R	CHIP_NAME_45	Chip name character 45	
[7:0]	R	CHIP_NAME_44	Chip name character 44	
RESERVED				
Address : C006_7040h				
GUID0 (GUID0)				
Address : C006_7044h				
[31:0]	R	GUID0	GUID 0	
GUID 1_2 (GUID1_2)				
Address : C006_7048h				
[31:16]	R	GUID2	GUID 2	
[15:0]	R	GUID1	GUID 1	
GUID 3_0 (GUID3_0)				
Address : C006_704Ch				
[31:24]	R	GUID3_3	GUID 3_3	
[23:16]	R	GUID3_2	GUID 3_2	
[15:8]	R	GUID3_1	GUID 3_1	
[7:0]	R	GUID3_0	GUID 3_0	
GUID 3_1 (GUID3_1)				
Address : C006_7050h				
[31:24]	R	GUID3_7	GUID 3_7	
[23:16]	R	GUID3_6	GUID 3_6	
[15:8]	R	GUID3_5	GUID 3_5	
[7:0]	R	GUID3_4	GUID 3_4	
ECID Control Register 0 (EC0)				
Address : C006_7054h				
[31:10]	-	RESERVED	Reserved	
[9:7]	R/W	A_FF	Programmable A	3b0

Bit	R/W	Symbol	Description	Reset Value
[6]	R/W	CS_FF	Programmable CS	1'b0
[5]	R/W	SIGDEV_FF	Programmable SIGDEV	1'b0
[4]	R/W	FSET_FF	Programmable FSET	1'b0
[3]	R/W	PRCHG_FF	Programmable PRCHG	1'b0
[2:0]	R	BONDING_ID	Boinding ID	
ECID Control Register 1 (EC1)				
Address : C006_7058h				
[31:3]	-	RESERVED	Reserved	
[2]	R/W	PROG_FF	Programmable PROG	1'b0
[1]	R/W	SCK_FF	Programmable SCK	1'b0
[0]	R/W	SDI_FF	Programmable SDI	1'b0
ECID Control Register 2 (EC2)				
Address : C006_705Ch				
[31:16]	-	RESERVED	Reserved	
[15]	R	HW_DONE	ECID initialize done	1'b0
[14:5]	-	RESERVED	Reserved	
[4]	R/W	HDCP_EFUSE_SEL	HDCP Key select 0 : Secure Boot 1 : Secure JTAG	1'b0
[3:2]	-	RESERVED	Reserved	
[1:0]	R/W	SEL_BANK	eFUSE select 0 : ECID 1 : Secure Boot 2 : Secure JTAG 3 : Backdoor JTAG	2'b0

13.4 Application Notes

- Permitting Over-the-cell routing. In chip-level layout, over-the-cell routing in EFROM_LP is permitted without any restrictions over Metal-4 layer.
- Incoming power bus should be adjusted to guarantee NOT more than 5% voltage drop at typical-case current levels.
- Reduce resistance ($<3\Omega$) between PAD and FSOURCE pin of EFROM_LP.
- Connect routing signal to all of the FSOURCE pins.

When using two or more e-fuse sets, FSOURCE line can be shared among e-fuse sets, but program-mode operation should be applied sequentially. (For example, at first, operate e-fuse0 and then operate e-fuse1 for program-mode) It applies same during sense-mode, but there is a guideline for multi-sensing, only sense-mode. Keep the below rules. Refer to integration guide document for further information.

Allowed maximum resistance can be obtained by $3\Omega / \# \text{ of } 128\text{bit efuse box}$.

If customer wants to run multi-sensing,

→ $3\Omega: 128\text{bit} * 1\text{ea}, 1.5\Omega: 128\text{bit} * 2\text{ea}, 1\Omega: 128\text{bit} * 3\text{ea}, 0.5\Omega: 128\text{bit} * 6\text{ea}$

Section 14. Memory Controller

14.1 Overview

The NXP4330D/Q Memory Controller is based on a Unified Memory Architecture (UMA). This Controller consists of two control units: MCU-A, MCU-S. Each unit has dedicated control pins.

14.1.1 Unified Memory Architecture (UMA)

- Two Separate Memory Controller:
 - MCU-A : DDR3/LPDDR3/LPDDR2
 - MCU-S : Static Memory
- MCU-A features:
 - MCU-A is organized DREX and DDRPHY
 - Supports DDR3/LPDDR3/LPDDR2 memory
 - Supports 8/16/32bit SDRAM of 2GByte
 - Single Bank of Memory(32bit data bus width)
 - Supports Power down mode
 - Supports Self Refresh mode
- MCU-S features:
 - Static memory
 - two Static Memory Chip Selects
 - NAND Flash Interface
 - 23-bit address supports using latch address
 - SLC NAND, MLC NAND with ECC(Supports BCH-algorithm)
 - Static Memory Map Shadow

14.1.2 Block Diagram

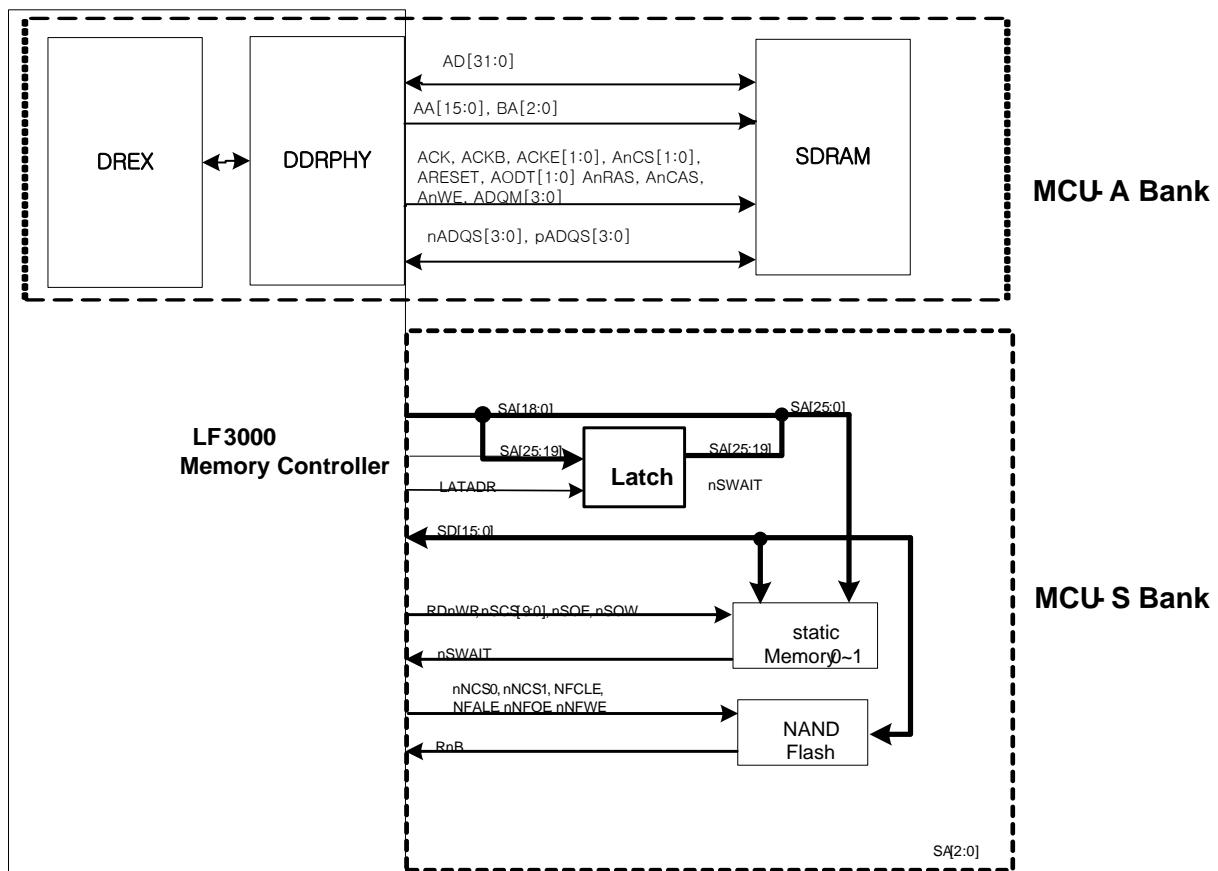


Figure 14-1. Memory Controller Block Diagram

14.2 Functional Descriptions

The memory controller area of the NXP4330D/Q is divided into an MCU-A and MCU-S bank. The MCU-A bank is connected to DDR3/LPDDR3/LPDDR2 which is the main memory of the NXP4330D/Q and 32-bit data bus width.

The MCU-S bank is the static bank and can be connected to Static Memory/Device, NAND (Refer to Figure 14-2 Memory Map)

14.2.1 MCU-A Bank Feature

- Compatible with JEDEC standard LPDDR2-S4/LPDDR3/DDR3 SDRAMs
- Supports 1:2 synchronous operation between bus clock and Memory clock
- Supports up to two memory ranks (chip selects) and 4/8 banks per memory chips
- Supports 512Mb, 1Gb, 2Gb, 4Gb and 8Gbit density per a chip select
- Supports outstanding exclusive accesses
- Supports bank selective precharge policy
- DREX Clock : MBCLK(400MHz)
- DDRPHY Clock : MCLK(800MHz), MDCLK(800MHz)
- MBCLK : MCLK = 1 : MBCLKx2
- MDCLK : MDCLK 0° phase Master DLL clock (400~800MHz). This clock should be the same frequency clock with clk2x in normal mode and generated from the same PLL which MCLK is using. But Master DLL is not able to lock under 400MHz. If MCLK is under 400MHz, the double frequency of clk2x can be used for locking Master DLL for the low frequency operation

14.2.2 MCU-S Bank Feature

- Latched Addressing

The number of pins that are connected to the outside is ADDR [18:0]. Since, however, the total address of the MCU-S Bank is 26-bit. ADDR[8:2] and ADDR[25:19] are allotted to the same pin, the system has a structure in which addresses are output two times. If the system uses ADDR[19] or more, the setting of higher address (ADDR[25:19]) is possible via System Configuration Pin (CfgSTLATADD). In this event, ADDR[25:19] which is configured by using external Latch IC first should be latched. (For more detailed information, refer to NXP4330D/Q Application Note – MCU-S.)

- 16-bit data bus width

Static memory register except NAND flash(8bits) has bus width select registers

- Static memory Controller

Normal static memory (SRAM and ROM) or static devices are connected.

Up to ten Static Chip Select signals exist.

- NAND Flash Controller

It supports both the small and large block NAND flash memories.

Up to four NAND flash memories can be connected.

Supports both SLC and MLC NAND flash memories.

Up to 4/8/16/24/40/60 bit error correction/(1024 or 512byte) using Binary-BCH coding

14.2.3 Memory Map

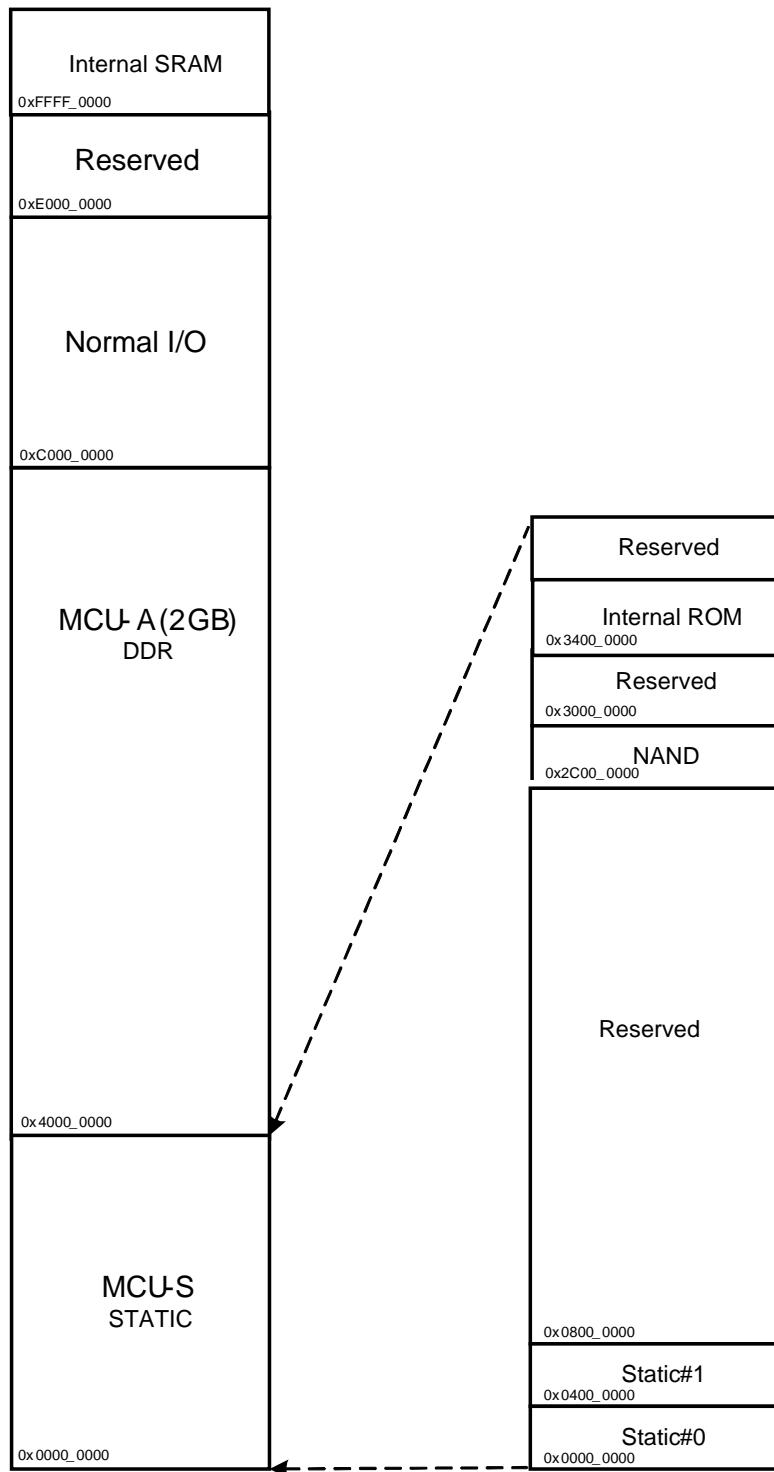


Figure 14-2. Memory Map

The memory map is roughly divided into one SDRAM Bank (MCU-A) and one static bank (MCU-S). The static bank consists of NAND Flash controller, Static Memory controller. The MCU-A bank consist of a Linear Array area and Display Array area.

14.2.4 MCU-A Address Mapping

DREX modifies the address of the AXI transaction coming from the AXI slave port into a memory address – chip select, bank address, row address, column address and memory data width.

To map chip select of memory device to a specific area of the address map, the **chip_base** and **chip_mask** bit-fields of the **MemConfig0** register needs to be set (Refer to Register Descriptions). If chip1 of the memory device exists, the **MemConfig1** register must also be set. Then, the AXI address requested by AXI Masters is divided into the AXI base address and AXI offset address. The AXI base address activates the appropriate memory chip select and the AXI offset address is mapped to a memory address according to the bank, row, column number, and data width set by the **MemConfig0/1** and **MemControl** register.

There are two ways to map the AXI offset address as shown below: 1.3.1 linear mapping 1.3.2 interleaved mapping

14.2.4.1 Linear mapping

As shown in Figure 14-3 the linear mapping method maps the AXI address in the order of bank, row, column and width. Since the bank address does not change for at least one bank size, applications that use linear address mapping have a high possibility to access the same bank.

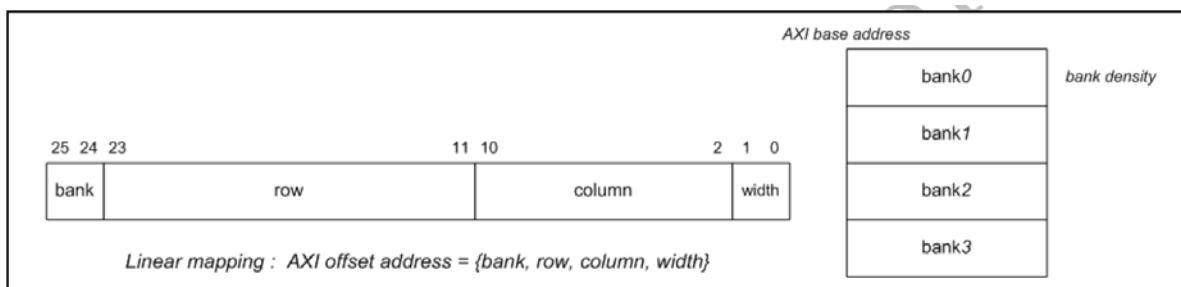


Figure 14-3. Linear Address Mapping

14.2.4.2 Interleaved mapping

As shown in Figure 14-4 the interleaved mapping method maps the AXI address in the order of row, bank, column and width. The difference between above two methods is that the bank and row order is different. For accesses beyond a row size, interleaved mapping accesses a different bank. Therefore, applications that use interleaved mapping access numerous banks. It causes better performance but more power consumptionCommand Arbiter

Command Arbiter re-arbitrates with the result of Fast Arbiter and Slow Arbiter. In this case, Fast Arbiter is always processed ahead with fixed method.

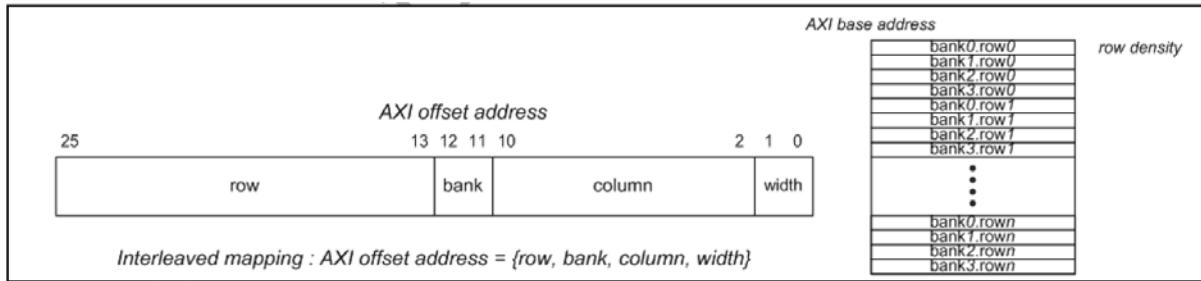


Figure 14-4. Linear Address Mapping

14.2.5 Low Power Operation

The controller executes a low power memory operation in five ways as described below. Each feature is independent of each other and executed at the same time.

14.2.5.1 AXI low power channel

The controller has an AXI low power channel interface to communicate with low power management units such as the system controller, which makes the memory device go into self refresh mode.

14.2.5.2 Dynamic power down

An SDRAM device has an active/precharge power down mode. This mode is triggered by deasserting CKE to LOW. When any of the banks is open, it enters active power down mode. Otherwise, it enters precharge power down mode.

When the request buffers remain empty for certain number of cycles (PwrDnConfig.dpwrDn_cyc register), DREX-1 changes the memory device's state to active/precharge power down automatically. The memory device enters Active/precharge power down mode or Forced precharge power down mode according to the SFR setting. The description of the two power down modes are as follows:

- 1) Active/precharge power down mode: Enter power down w/o considering whether there is a row open or not.
- 2) Forced precharge power down mode: Enter power down after closing all banks.

When DREX-1 receives a new AXI transaction while memory device is in power down mode, it automatically wakes up the memory device from power down state and executes in a normal operation state.

14.2.5.3 Dynamic self refresh

Similarly to the dynamic power down feature, if the request buffers remain empty for certain number of cycles (PwrDnConfig.dsref_cyc register), DREX-1 changes the memory device's state to self-refresh mode. Since exiting power down mode requires many cycles, a longer idle cycle threshold is recommended for dynamic self-refresh entry than the threshold for dynamic power down.

14.2.5.4 Clock stop

To reduce the I/O power of the memory device and the controller, it is possible to stop the clock if the LPDDR / LPDDR2-S4 is in idle mode, or self refresh mode and DDR2/DDR3 is in self refresh mode. If this feature is enabled, the controller automatically executes the clock stop feature. In DDR3, clock stop feature must be turn-on and off considering tCKSRX/tCKSRE/tCKESR timing by software.

14.2.5.5 Direct command

Use the direct command feature to send a memory command directly to the memory device through the APB3 port. This way, it is possible to force the memory device to enter active/precharge power down, self-refresh or deep power down mode

14.2.6 MCU-A application Note

14.2.6.1 DREX initialization

LPDDR2/3

The following sequence should be used to initialize LPDDR2 devices. Unless specified otherwise, these steps are mandatory.

- 1) To provide stable power for memory device, DREX-1 must assert and hold CKE to a logic low level. Then apply stable clock.
- 2) Set the right value to PHY control register0 for LPDDR2/3 operation mode. If read leveling is needed, check LPDDR2/3 IO calibration MRR data and match it to PHY control register1's ctrl_rlvl_rdata_adj field. (Refer to PHY manual)
- 3) Set the PHY for dqs pulldown mode. (Refer to PHY manual, PHY control register 14)
- 4) Set the ConControl. At this moment, assert the dfi_init_start field to high but the aref_en field should be off.
- 5) Wait for the PhyStatus0.dfi_init_complete field to change to [1].
- 6) Set the ConControl. At this moment, deassert the dfi_init_start field to low and the aref_en field should be off.
- 7) Set the PhyControl0.fp_resync bit-field to [1] to update DLL information for at least one pclk cycle.
- 8) Set the PhyControl0.fp_resync bit-field to [0].
- 9) Set the MemControl and PhyControl0. At this moment, all power down modes including sl_dll_dyn_con should be off.
- 10) Set the MemBaseConfig0 register. If there are two external memory chips, set the MemBaseConfig1 register.
- 11) Set the MemConfig0 register. If there are two external memory chips, also set the MemConfig1 register.
- 12) Set the PrechConfig and PwrDnConfig registers.
- 13) Set the TimingAref, TimingRow, TimingData and TimingPower registers according to memory AC parameters.
- 14) If QoS scheme is required, set the QosControl0~15 and QosConfig0~15 registers.
- 15) Set the PHY for LPDDR2/3 initialization. LPDDR2/3 initialization clock period is 18ns ~ 100ns. If LPDDR2/3 initialization clock period is 20ns, then follow below steps 16 ~ 22(refer to PHY manual).
- 16) Set the PHY ctrl_offsetr0~3 and ctrl_offsetw0~3 value to 0x7F
- 17) Set the PHY ctrl_offsetd value to 0x7F.
- 18) Set the PHY ctrl_force value to 0x7F.
- 19) Set the PHY ctrl_dll_on to low.
- 20) Wait for 10 PCLK cycles.
- 21) Set the PhyControl0.fp_resync bit-field to [1] to update DLL information.
- 22) Set the PhyControl0.fp_resync bit-field to [0].

- 23) Confirm that CKE has been as a logic low level at least 100ns after power on
- 24) Issue a NOP command using the DirectCmd register to assert and to hold CKE to a logic high level.
- 25) Wait for minimum 200us.
- 26) Issue a MRS command using the DirectCmd register to reset memory devices and program the operating parameters.
- 27) Wait for minimum 10us.
- 28) Issue proper lpddr2 initialization commands according to specification such as IO calibration (MR #10), device feature1,2 (MR #1, #2). Refer to LPDDR2/3 specification for details.
- 29) If there are two external memory chips, perform steps 24 ~ 28 for chip1 memory device.
- 30) Set the PHY ctrl_offsetr0~3 and ctrl_offsetw0~3 value to 0x0
- 31) Set the PHY ctrl_offsetd value to 0x0
- 32) Set the PHY ctrl_dll_on enable
- 33) Wait for 10 PCLK cycles.
- 34) Set the PHY ctrl_start value to '0'.
- 35) Set the PHY ctrl_start value to '1'.
- 36) Wait for 10 PCLK cycles.
- 37) Wait for the PhyStatus0.dfi_init_complete field to change to ?1?.
- 38) Set the PhyControl0.fp_resync bit-field to ?1? to update DLL information.
- 39) Set the PhyControl0.fp_resync bit-field to ?0?.
- 40) If any leveling/training is needed, disable ctrl_dll_on and set ctrl_force value. (Refer to PHY manual)
- 41) If write leveling for LPDDR3 is not needed, skip this procedure. If write leveling is needed, set LPDDR3 into write leveling mode using MRS command, set ODT pin high and tWLO using WRLVL_CONFIG0 regis-ter(offset=0x120) and set the related PHY SFR fields through PHY APB Interface (Refer to PHY manual). To generate 1 cycle pulse of dfi_wrdtdata_en_p0, write 0x1 to WRLVL_CONFIG1 register (Offset addr=0x124). To read the value of memory data, use CTRL_IO_RDATA(offset = 0x150). If write leveling is finished, then set ODT pin low and disable write leveling mode of LPDDR3
- 42) If CA calibration for LPDDR3 is not needed, skip this procedure. If CA calibration is needed, set the related PHY SFR fields through PHY APB Interface (Refer to PHY manual). Set LPDDR3 into CA calibration mode through MR41. For CKE assertion, deassertion, CA value and tADDR setting, use CACAL_CONFIG0(offset = 0x160). For Generation 1 cycle pulse of dfi_csn_p0, use CACAL_CONFIG1(offset = 0x164). To read the value of memory data, use CTRL_IO_RDATA_CH0/CH1 (offset = 0x150, 0x154). Note that CKE pin should be as-serted when MR41/48/42 command is issued and CKE pin should be deasserted during CA calibration. Also note that because CA SDLL code updating needs some cycles to complete, 10 PCLK cycles are needed be-fore issuing next command.
- 43) If read leveling is not needed, skip 44 ~ 48 and set proper value to PHY control register #2. If read leveling is needed, set the related PHY SFR fields through PHY APB Interface. The related register is PHY control regis-ter #1 and #2(mpr value, ctrl_rdlvl_gate_en and ctrl_rdlvl_en register, refer to PHY manual).
- 44) Set the RdlvlConfig.ctrl_rdlvl_data_en bit-field to 1'b1. Gate training is not supported.
- 45) Wait for the PhyStatus0.read_level_complete field to change to [1].
- 46) Disable the RdlvlConfig.ctrl_rdlvl_gate_en and ctrl_rdlvl_data_en.
- 47) Set the PhyControl0.fp_resync bit-field to [1] to update DLL information.

- 48) Set the PhyControl0.fp_resync bit-field to [0].
- 49) If write training is not needed, skip 50 ~ 55. If write training is needed, set the related PHY SFR fields through PHY APB Interface. The related register is PHY control register #2 and #26, refer to PHY manual)
- 50) Set write latency of PHY control register #26.
- 51) Enable WrtraConfig.write_training_en to issue ACT command. Refer to this register definition for row and bank address.
- 52) Wait for 10 PCLK cycles.
- 53) Enable write de-skewing of PHY control register #2.
- 54) Wait for the PhyStatus0.read_level_complete field to change to ?1?.
- 55) Disable write de-skewing of PHY control register #2.
- 56) After all leveling/training are completed, enable ctrl_dll_on. (Refer to PHY manual)
- 57) Set the PhyControl0.fp_resync bit-field to [1] to update DLL information.
- 58) Set the PhyControl0.fp_resync bit-field to [0].
- 59) Disable PHY gating control through PHY APB Interface (ctrl_atgate, see PHY manual).
- 60) If power down modes are required, set the MemControl register.
- 61) Set the ConControl to turn on an auto refresh counter.

DDR3

The following sequence should be used to initialize DDR3 devices. Unless specified otherwise, these steps are mandatory.

- 1) Apply power. RESET# pin of memory needs to be maintained for minimum 200us with stable power. CKE is pulled "Low" anytime before RESET# being de-asserted (min. time 10ns)
- 2) Set the PHY for DDR3 operation mode. . If read leveling is needed, check DDR3 MPR data and match it to PHY control register1's ctrl_rlvl_rdata_adj field. (Refer to PHY manual)
- 3) Set the PHY for dqs pulldown mode. (Refer to PHY manual, PHY control register 14)
- 4) If on die termination is required, enable PhyControl0.mem_term_en, PhyControl0.phy_term_en.
- 5) Set the ConControl. At this moment, assert the dfi_init_start field to high but the aref_en field should be off.
- 6) Wait for the PhyStatus0.dfi_init_complete field to change to [1].
- 7) Set the ConControl. At this moment, deassert the dfi_init_start field to low and the aref_en field should be off.
- 8) Set the PhyControl0.fp_resync bit-field to [1] to update DLL information.
- 9) Set the PhyControl0.fp_resync bit-field to [0].
- 10) Set the MemControl and PhyControl0. At this moment, all power down modes including sl_dll_dyn_con and periodic ZQ(pzq_en) should be off.
- 11) Set the MemBaseConfig0 register. If there are two external memory chips, set the MemBaseConfig1 register.
- 12) Set the MemConfig0 register. If there are two external memory chips, also set the MemConfig1 register.
- 13) Set the PrechConfig and PwrdnConfig registers.
- 14) Set the TimingAref, TimingRow, TimingData and TimingPower registers according to memory AC parameters.

- 15) If QoS scheme is required, set the QosControl0~15 and QosConfig0~15 registers.
- 16) Confirm that after RESET# is de-asserted, 500 us have passed before CKE becomes active.
- 17) Confirm that clocks(CK, CK#) need to be started and stabilized for at least 10 ns or 5 tCK (which is larger) before CKE goes active.
- 18) Issue a NOP command using the DirectCmd register to assert and to hold CKE to a logic high level.
- 19) Wait for tXPR(max(5nCK,tRFC(min)+10ns)) or set tXP to tXPR value before step 17. If the system set tXP to tXPR, then the system must set tXP to proper value before normal memory operation.
- 20) Issue an EMRS2 command using the DirectCmd register to program the operating parameters. Dynamic ODT should be disabled. A10 and A9 should be low.
- 21) Issue an EMRS3 command using the DirectCmd register to program the operating parameters.
- 22) Issue an EMRS command using the DirectCmd register to enable the memory DLL.
- 23) Issue a MRS command using the DirectCmd register to reset the memory DLL.
- 24) Issues a MRS command using the DirectCmd register to program the operating parameters without resetting the memory DLL.
- 25) Issues a ZQINIT commands using the DirectCmd register.
- 26) If there are two external memory chips, perform steps 18 ~ 25 procedures for chip1 memory device.
- 27) If any leveling/training is needed, disable ctrl_dll_on and set ctrl_force value. (Refer to PHY manual)
- 28) If write leveling is not needed, skip this procedure. If write leveling is needed, set DDR3 into write leveling mode using MRS command, set ODT pin high and tWLO using WRLVL_CONFIG0 register(offset=0x120) and set the related PHY SFR fields through PHY APB Interface (Refer to PHY manual). To generate 1 cycle pulse of dfl_wrdtdata_en_p0, write 0x1 to WRLVL_CONFIG1 register (Offset addr=0x124). If write leveling is finished, then set ODT pin low and disable write leveling mode of DDR3
- 29) If gate leveling is not needed, skip 30 ~ 33. Gate leveling is only supported DDR3 over 667MHz. If gate leveling is needed, set the related PHY SFR fields through PHY APB Interface. The related register is PHY control register #0, #1 and #2 (Refer to PHY manual)
- 30) Set the RdlvlConfig.ctrl_rdlvl_gate_en bit-field to 1'b1.
- 31) Wait for the PhyStatus0.read_level_complete field to change to [1].
- 32) Disable DQS pulldown mode.(Refer to PHY manual)
- 33) Disable the RdlvlConfig.ctrl_rdlvl_gate_en and ctrl_rdlvl_data_en.
- 34) If read leveling is not needed, skip 35 ~ 39 and set proper value to PHY control register #2. If read leveling is needed, set the related PHY SFR fields through PHY APB Interface. The related register is PHY control register #1 and #2 (mpr value, ctrl_rdlvl_gate_en and ctrl_rdlvl_en register, refer to PHY manual).
- 35) Set the ctrl_rdlvl_data_en bit-field to 1'b1.
- 36) Wait for the PhyStatus0.read_level_complete field to change to [1]
- 37) Disable the RdlvlConfig.ctrl_rdlvl_gate_en and ctrl_rdlvl_data_en.
- 38) Set the PhyControl0.fp_resync bit-field to [1] to update DLL information.
- 39) Set the PhyControl0.fp_resync bit-field to [0].
- 40) If write training is not needed, skip 41 ~ 46. If write training is needed, set the related PHY SFR fields through PHY APB Interface. The related register is PHY control register #2 and #26, refer to PHY manual)

- 41) Set write latency of PHY control register #26.
- 42) Enable WrtraConfig.write_training_en to issue ACT command. Refer to this register definition for row and bank address.
- 43) Wait for 10 PCLK cycles.
- 44) Enable write de-skewing of PHY control register #2.
- 45) Wait for the PhyStatus0.read_level_complete field to change to [1].
- 46) Disable write de-skewing of PHY control register #2.
- 47) After all leveling/training are completed, enable ctrl_dll_on. (Refer to PHY manual)
- 48) Set the PhyControl0.fp_resync bit-field to [1] to update DLL information.
- 49) Set the PhyControl0.fp_resync bit-field to [0].
- 50) Disable PHY gating control through PHY APB Interface (ctrl_atgate, refer to PHY manual).
- 51) If power down modes or periodic ZQ (pzq_en) are required, set the MemControl register.
- 52) Set the ConControl to turn on an auto refresh counter.

14.2.6.2 DDRPHY initialization

After power-up and system PLL locking time, system reset(rst_n) is released.

- 1) Select Memory Type (=PHY_CON0[12:11]).
 - d. ctrl_ddr_mode=2'b11 (LPDDR3)
 - e. ctrl_ddr_mode=2'b10 (LDLR2)
 - f. ctrl_ddr_mode=2'b01 (DDR3)

NOTE: If ctrl_ddr_mode[1]=1'b1, lpddr2_cmd=14'h000E(=PHY_CON25[13:0]), cmd_default=14'h000F(=PHY_CON26[13:0])
- 2) Set Read Latency(RL), Burst Length(BL) and Write Latency(WL)
 - a. Set RL in PHY_CON42[4:0].
 - b. Set BL in PHY_CON42[12:8].
 - c. Set WL in PHY_CON26[20:16].
- 3) ZQ Calibration(Please refer to "8.5 ZQ I/O CONTROL PROCEDURE" for more details)
 - a. Set Drive Strength (=zq_mode_dds or PHY_CON39[27:0]) properly (Please refer to p56, p63).- Please don't use default value(=0x0)
 - b. Enable and Disable "zq_clk_div_en" in PHY_CON16[18]
 - c. Enable "zq_manual_str" in PHY_CON16[1]
 - d. Wait until "zq_cal_done"(=PHY_CON17[0]) is enabled.
 - e. Disable "zq_manual_str"(=PHY_CON16[1])
- 4) Memory Controller should assert "dfi_init_start" from LOW to HIGH.
- 5) Memory Controller should wait until "dfi_init_complete" is set
 - a. DLL lock will be processed.
 - b. If the frequency of "MDCLK" is changed during operation, "ctrl_start" should be clear and set to lock again.

- 6) Enable DQS pull down mode
- 7) Memory Controller should assert "dfi_ctrlupd_req" after "dfi_init_complete" is set.
- 8) Start Memory Initialization.
- 9) Skip the following steps if Leveling and Training are not required.
 - a. Constraints during Leveling
 - Support BL=4 or 8 during Leveling. (Don't use BL=16)
 - Not support Memory ODT(On-Die-Termination) during Write DQ Calibration.
 - b. Enable "ctrl_atgate" in PHY_CON0[6].
 - c. Enable "p0_cmd_en" in PHY_CON0[14].
 - d. Enable "InitDeskewEn" in PHY_CON2[6].
 - e. Enable "byte_rdlvl_en" in PHY_CON0[13].
 - f. Set "rdlvl_pass_adj=4'h6" in PHY_CON1[19:16].
 - g. Set "ddr3_cmd=14'h105E" as default value (=PHY_CON25[29:16])
 - h. Set "lpddr2_cmd=14'h107E" as default value (=PHY_CON25[13:0])
 - i. Set "cmd_default= 16'h000F(LPDDR2, LPDDR3), 16'h107F(DDR2, DDR3)" as default value (=PHY_CON26[13:0])
 - j. Recommend that "rdlvl_incr_adj=7'h01" for the best margin.
 - Calibration time can be shorter by adjusting "rdlvl_incr_adj" in PHY_CON2[22:16].
 - k. Disable "ctrl_dll_on" in PHY_CON12[5] before Leveling.
 - Read "ctrl_lock_value[8:2]" in PHY_CON13[16:10].
 - Update "ctrl_force[6:0]" in PHY_CON12[14:8] by the value of "ctrl_lock_value[9:2]".
 - l. Write Leveling (refer to 8.1.1)
 - m. CA Calibration(refer to 8.1.2)
 - n. Gate Leveling (refer to 8.1.3)
 - It should be used only for DDR3 (800MHz). Please don't use under 800MHz.
 - o. Read DQ Calibration(=Read Leveling) (refer to 8.1.4)
 - p. After Read DQ Calibration, refer to "T_rddata_en" to know where "dfi_rddata_en_p0/p1" is enabled.
 - Read "T_rddata_en" timing parameters in PHY_CON18 after Read DQ Calibration.
 - q. Write DQ Calibration (refer to 8.1.5)
 - r. Enable "ctrl_dll_on" in PHY_CON12[5].
 - s. Disable "ctrl_atgate" in PHY_CON0[6] if controller controls "ctrl_gate_p0/p1" and "ctrl_read_p0/p1" directly.
 - t. Enable "DLLDeskewEn" (=PHY_CON2[12]) to compensate Voltage, Temperature variation during operation.
- 10) Controller should assert "dfi_ctrlupd_req" to make sure All SDLL is updated.
 - a. If "ca_swap_mode" is enabled, please don't use "ctrl_atgate=1" during normal operation.

NOTE: The goal of data eye training (=Read, Write DQ Calibration) is to identify the delay at which the read DQS rising edge aligns with the beginning and end transitions of the associated DQ data eye. By identifying these delays, the system can calculate the midpoint between the delays and accurately center the read DQS within the DQ data eye.

WRITE LEVELING

Write Leveling compensates for the additional flight time skew delay introduced by the package, board and on-chip with respect to strobe(=DQS) and clock. The flight time skew between DQS and clock should be under 240ps to be compensated properly and suppose that the clock will be delayed than DQS

- 1) Memory Controller should configure Memory (DDR3, LPDDR3) in Write Level mode.
 - a. Memory Controller should assert "ODT[1:0]" signals(=dfi_odt_p0/p1) during Write Leveling.
- 2) Configure PHY in Write Level mode.
 - a. Enable "wrlvl_mode" in PHY_CON0[16].
 - b. "NOP"(CS HIGH at the clock rising edge N) should be used during "wrlvl_mode"=1(Refer to p105).
- 3) To find out the optimal Write Level De-skew DLL code for the alignment between CK and DQS
 - a. Set each DLL code (PHY_CON30[6:0], PHY_CON[14:8], PHY_CON[23:17], PHY_CON[30:24]).(1)
 - The start code value should be 0x8. (0x8~0x38)
 - PHY_CON30[6:0] (= "DQS[0]" SDLL code)
 - PHY_CON30[14:8] (= "DQS[1]" SDLL code)
 - PHY_CON30[23:17] (= "DQS[2]" SDLL code)
 - PHY_CON30[30:24] (= "DQS[3]" SDLL code)
 - b. Update SDLL code(PHY_CON30[16]).(2)
 - Enable "ctrl_wrlvl_resync" in PHY_CON30[16]
 - Disable " ctrl_wrlvl_resync" in PHY_CON30[16]
 - c. Memory Controller should generate 1 cycle pulse of "dfi_wrdata_en_p0".(3)
 - d. Memory Controller should read the value of "ctrl_io_rdata[0]" which is output of PHY.(4)
 - If it is zero, Increment "DQS[0]" SDLL code by "1" and then go to "2" to update "DQS[0]" SDLL code.
 - If it is one, the previous "DQS[0]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.
 - e. Memory Controller should read the value of "ctrl_io_rdata[8]" which is output of PHY.(5)
 - If it is zero, Increment "DQS[1]" SDLL code by "1" and then go to "2" to update "DQS[1]" SDLL code.
 - If it is one, the previous "DQS[1]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.
 - f. Memory Controller should read the value of "ctrl_io_rdata[16]" which is output of PHY.(6)
 - If it is zero, Increment "DQS[2]" SDLL code by "1" and then go to "2" to update "DQS[2]" SDLL code.
 - If it is one, the previous "DQS[2]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.
 - g. Memory Controller should read the value of "ctrl_io_rdata[31]" which is output of PHY.(7)
 - If it is zero, Increment "DQS[3]" SDLL code by "1" and then go to "2" to update "DQS[3]" SDLL code.

- If it is one, the previous "DQS[3]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.

4) Configure PHY in normal mode after 4~7 are finished.(8)

- a. Disable "wrlvl_mode" in PHY_CON0[16].

5) Memory Controller should configure Memory (DDR3, LPDDR3) in normal mode.(9)

CA CALIBRATION

1) Controller should configure Memory (LPDDR3) in CA Calibration mode.

2) Configure PHY in CA Calibration mode.

- a. Enable "wrlvl_mode" in PHY_CON0[16].
- b. Enable "ca_cal_mode" in PHY_CON2[23].

3) How to find the optimal CA SDLL code. (=PHY_CON10[7:0])

- a. Change CA SDLL code in PHY_CON10[7:0]. (1)
 - The start code value should be 0x8.
- b. Update CA SDLL code in PHY_CON10[7:0]. (2)
 - Enable "ctrl_resync" in PHY_CON10[24]
 - Disable "ctrl_resync" in PHY_CON10[24]
- c. CA to DQ mapping change to calibrate CA[3:0], CA[8:5]. (3)
 - Mode Register Write to MR#41 by Controller.
 - Memory Controller should disable "dfi_cke_p0" and "dfi_cke_p1". (dfi_cke_p0/p1=0)
- d. Memory Controller should generate 1 cycle pulse of "dfi_cs_n_p0" with "dfi_address_p0 = 20'h3FF".
- e. Memory Controller should read and save the value of "ctrl_io_rddata[15:0]" which is output of PHY.
 - CA[3:0] at rising edge CK(=CA_L[3:0]) is equal to
 - CA[8:5] at rising edge CK(=CA_L[8:5]) is equal to {ctrl_io_rdata[14], ctrl_io_rdata[12], ctrl_io_rdata[10], ctrl_io_rdata[8]}.
 - CA[3:0] at falling edge CK(=CA_H[3:0]) is equal to {ctrl_io_rdata[7], ctrl_io_rdata[5], ctrl_io_rdata[3], ctrl_io_rdata[1]}.
 - CA[8:5] at falling edge CK(=CA_H[8:5]) is equal to {ctrl_io_rdata[15], ctrl_io_rdata[13], ctrl_io_rdata[11], ctrl_io_rdata[9]}.
- f. CA to DQ mapping change to calibrate CA[4], CA[9]. (4)
 - Memory Controller should enable "dfi_cke_p0" and "dfi_cke_p1". (dfi_cke_p0/p1=1)
 - Mode Register Write to MR#48 by Controller.
 - Memory Controller should disable "dfi_cke_p0" and "dfi_cke_p1". (dfi_cke_p0/p1=0)
- g. Memory Controller should generate 1 cycle pulse of "dfi_cs_n_p0" with "dfi_address_p0 = 20'h3FF".
- h. Memory Controller read and save the value of "ctrl_io_rddata[1:0]" and "ctrl_io_rddata[8:9]" which is output of PHY.
 - CA[4] at rising edge CK(=CA_L[4]) is equal to "ctrl_io_rddata[0]"
 - CA[9] at rising edge CK(=CA_L[9]) is equal to "ctrl_io_rddata[8]"
 - CA[4] at falling edge CK(=CA_H[4]) is equal to "ctrl_io_rddata[1]"

- CA[9] at falling edge CK(=CA_H[9]) is equal to "ctrl_io_rddata[9]"
 - i. Check if "CA_L = 10'h3FF" and "CA_H = 10'h000" or not. (5)
 - j. If not equaled,
 - Go to "6" until it searches for the leftmost code value. (7)
 - If it already saved the leftmost code value, save the current SDLL code by the rightmost code value (=VWMR). Go to "11". (10)
 - k. If equaled,
 - If it is matched for the first time, save the current SDLL code by the leftmost code value(=VWML).
 - Go to "6". (8)
 - Go to "6" until it searches for the rightmost code value. (9)
 - l. Increment SDLL code by "1" and then go to "2" to update SDLL code. (6)
- 4) Calculate the optimal CA SDLL code(=PHY_CON10[7:0]). (11)
- a. Calculate the optimal CA SDLL code(=VWMC) by the following formula.
 - b. VWMC = VWML + (VWMR - VWML)/2
 - c. Update CA SDLL code by using "VWMC".
- 5) Configure PHY in normal mode.
- a. Disable "wrlvl_mode" in PHY_CON0[16].
- 6) Memory Controller should configure Memory (LPDDR3) in normal mode.

GATE LEVELING

The goal of gate training is to locate the delay at which the initial read DQS rising edge aligns with the rising edge of the read DQS gate(=ctrl_gate_p0/p1). Once this point is identified, the read DQS gate can be adjusted prior to the DQS, to the approximate midpoint of the read DQS preamble. You can use "GATE Leveling" when using "DDR3 memory" over 800MHz.

- 1) Controller should configure Memory (DDR3) in MPR mode. (Please refer to JEDEC Standard.)
- 2) Set "lpddr2_addr=20'h208"(=PHY_CON22[19:0]) to issue MR32 during GATE Leveling (LPDDR2/3)
 - a. In case of CA swap mode, lpddr2_addr=20'h041 to issue MR32 during GATE Leveling
- 3) Set Gate Leveling Mode.(1)
 - a. Enable "gate_cal_mode" in PHY_CON2[24]
 - b. Enable "ctrl_shgate" in PHY_CON0[8]
 - c. Set "ctrl_gateduradj[3:0] (=PHY_CON1[23:20]) in the following way.
 - 4'b0000" (DDR3, DDR2)
 - 4'b1011" (LPDDR3)
 - 4'b1001" (LPDDR2)
- 4) Memory Controller should assert "dfi_rdlvl_en" and "dfi_rdlvl_gate_en" to do read leveling.(2)
- 5) Memory Controller should wait until "dfi_rdlvl_resp" is set.(3)
 - a. The maximum waiting time will be 20us. If the any command (the refresh or pre-charge command) is required within 20us, please issue those commands before "(1)".
- 6) Memory Controller should deassert "dfi_rdlvl_en" and "dfi_rdlvl_gate_en" after "dfi_rdlvl_resp" is disabled.
- 7) Disable DQS pull down mode.(4)

8) Memory Controller should configure Memory (DDR3) in normal mode.

READ DQ CALIBRATION (=READ LEVELING, READ DESKEWING)

Read DQ Calibration adjusts for the delays introduced by the package, board and on-chip that impact the read cycle.

1) In case of using DDR3 Memory,

- a. Memory Controller should configure Memory in MPR mode. (Please refer to JEDEC Standard)
- b. Set "PHY_CON1[15:0]" by "0xFF00" if "Pre-defined Data Pattern" is "[0x0000_0000,0x0101_0101, 0x0000_0000,0x0101_0101]" (byte_rdlvl_en=1)
- c. Set "PHY_CON1[15:0]" by "0x0100" if "Pre-defined Data Pattern" is "[0x0000_0000,0xFFFF_FFFF, 0x0000_0000,0xFFFF_FFFF]" in MPR mode. (byte_rdlvl_en=1)

NOTE: Read DQ Calibration with "byte_rdlvl_en=0" should be done after Write DQ Calibration. Set "PHY_CON1[15:0]" by "0xFF00". "MPR Data Pattern" should be "[0x0000_0000,0xFFFF_FFFF, 0x0000_0000,0xFFFF_FFFF]".

2) In case of using LPDDR3 or LPDDR2 Memory,

- a. Set "PHY_CON1[15:0]" by "0x00FF" if "MRR32 DQ Pattern" is "[0x0101_0101,0x0000_0000, 0x0101_0101, 0x0000_0000]" (byte_rdlvl_en=1)
- b. Set "PHY_CON1[15:0]" by "0x0001" if "MRR32 DQ Pattern" is "[0xFFFF_FFFF,0x0000_0000, 0xFFFF_FFFF, 0x0000_0000]". (byte_rdlvl_en=1)

NOTE: Read DQ Calibration with "byte_rdlvl_en=0" should be done after Write DQ Calibration. Set "PHY_CON1[15:0]" by "0x00FF". "MRR32 DQ Pattern" should be "[0xFFFF_FFFF,0x0000_0000, 0xFFFF_FFFF, 0x0000_0000]".

- c. Set "lpddr2_addr=20'h208" (=PHY_CON22[19:0]) to issue MR32 during Calibration (LPDDR2/3)
 - In case of CA swap mode, lpddr2_addr=20'h041 to issue MR32 during Calibration.

3) Set Read Leveling Mode.(1)

- a. Enable "rd_cal_mode" in PHY_CON2[25]

4) Memory Controller should assert "dfi_rdlvl_en" to do read leveling.(2)

5) Memory Controller should wait until "dfi_rdlvl_resp" is set.(3)

- a. The maximum waiting time will be 20us. If the any command (the refresh or pre-charge command) is required within 20us, please issue those commands before "(1)".

6) Memory Controller should deassert "dfi_rdlvl_en" after "dfi_rdlvl_resp" is enabled.(4)

7) Memory Controller should configure Memory (DDR3) in normal mode.

WRITE DQ CALIBRATION (=WRITE DESKEWING)

Write DQ Calibration adjusts for the delays introduced by the package, board and on-chip that impact the write cycle.

1) Set Write Latency(WL) before Write Training(1)

- a. Set "T_wrdata_en" by "WL" in PHY_CON26[20:16].
 - WL is defined as clock cycles between the write command and the first valid rising edge of DQS

2) Memory Controller should issue "Active Command".

3) PHY will keep writing and reading the pattern in "PHY_CON1[15:0]" for Write DQ Calibration according to the

following settings.(2)

- a. The column address should be defined in "lpddr2_addr" in PHY_CON22[19:0](LPDDR2, LPDDR3).
 - "lpddr2_addr[9:0]" correspond to CA[9:0] at the rising edge of CK, and "lpddr2_addr[19:10]" to CA[19:10] at the falling edge of CK.
 - It should be the "READ" command. For example, lpddr2_addr=20'h5 if the column address is 11'h0 and bank address is 3'b000. In case of CA swap mode, lpddr2_addr=20'h204 if the column address is 11'h0 and bank address is 3'b000.
- b. The column address should be defined in "ddr3_addr" in PHY_CON24[15:0] (DDR3)
 - For example, if the column address (=ddr3_addr) is "0x0", PHY will write and read the pre-defined pattern (=PHY_CON1[15:0]) at 0x0, 0x4, 0x8 and 0xC for DQ Calibration. (BL=4)
- c. In case of using LPDDR2 or LPDDR3 memory,
 - Set "PHY_CON1[15:0]=0x0001" and "byte_rdlvl_en=1"(=PHY_CON0[13])
 - Set "PHY_CON1[15:0]=0x00FF" and "byte_rdlvl_en=0"(=PHY_CON0[13]) for Deskewing.
- d. In case of using DDR3 memory,
 - Set "PHY_CON1[15:0]=0x0100" and "byte_rdlvl_en=1"(=PHY_CON0[13])
 - Set "PHY_CON1[15:0]=0xFF00" and "byte_rdlvl_en=0"(=PHY_CON0[13]) for Deskewing.

4) Enable "p0_cmd_en" in PHY_CON[14].

5) Set Write Training Mode(3)

- a. Enable "wr_cal_mode" in PHY_CON2[26].

6) Enable "wr_cal_start" in PHY_CON2[27] to do Write DQ Calibration.(4)

7) Memory Controller should wait until "dfi_rdlvl_resp" is set.(5)

- a. The maximum waiting time will be 50us. If the any command (the refresh or pre-charge command) is required within 50us, please issue those commands before "(3)".

8) Disable "wr_cal_start" in PHY_CON2[27] after "dfi_rdlvl_resp" is enabled.(6)

14.2.7 DLL Lock and ZQ I/O Calibration

DLL Lock and ZQ I/O calibration should be done prior to the initialization of MCUA Bank Memory following System Power on.

14.2.7.1 Static Memory Map Shadow

Figure 14-5 Static Memory is composed of Static#0~#1(External Static Memory), Static#13(Internal ROM) and NAND as Figure.

However, base address of internal ROM, and nSCS[0] is changed according to system boot mode. In internal ROM Boot Mode, base address of internal ROM is supposed to be exchanged with that of nSCS[0].

The previous Base Address remains in case of External Static Boot.

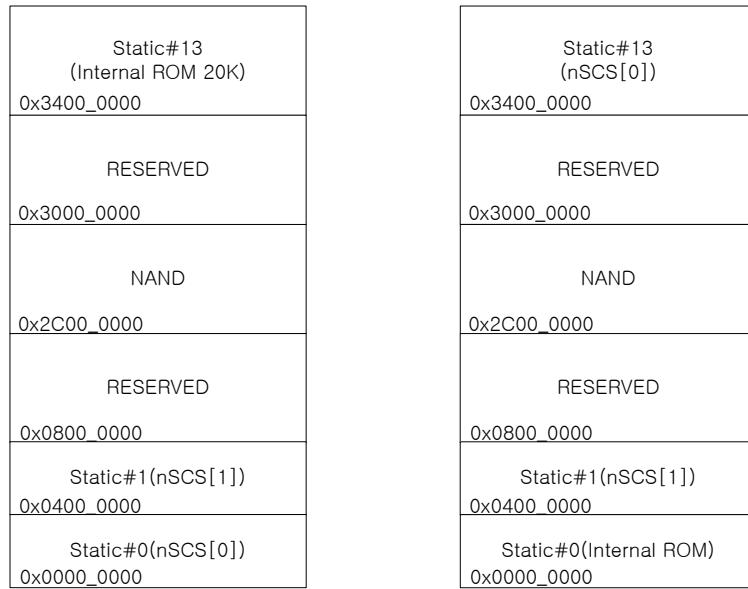


Figure 14-5. Static Memory Map Shadow

14.2.7.2 Interface

32bit Data bus width SDRAM Interface of MCU-A Bank

MCU-A supports 32bit data buswidth to CS[0] and CS[1] respectively.

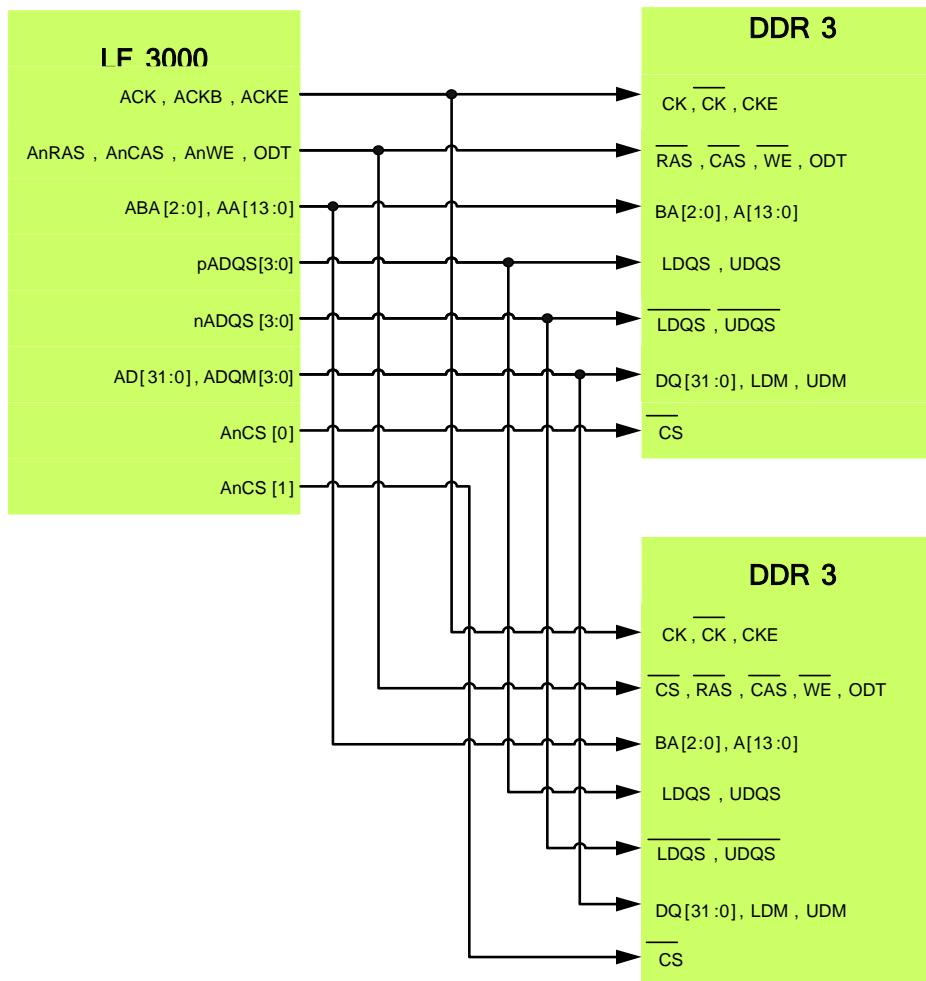


Figure 14-6. 16bit Data bus width SDRAM Interface

16bit Data bus width SDRAM Interface of MCU-A Bank

16bit DDR3 SDRAM can add each two Memory to CS[0] and CS[1] respectively, which total four.

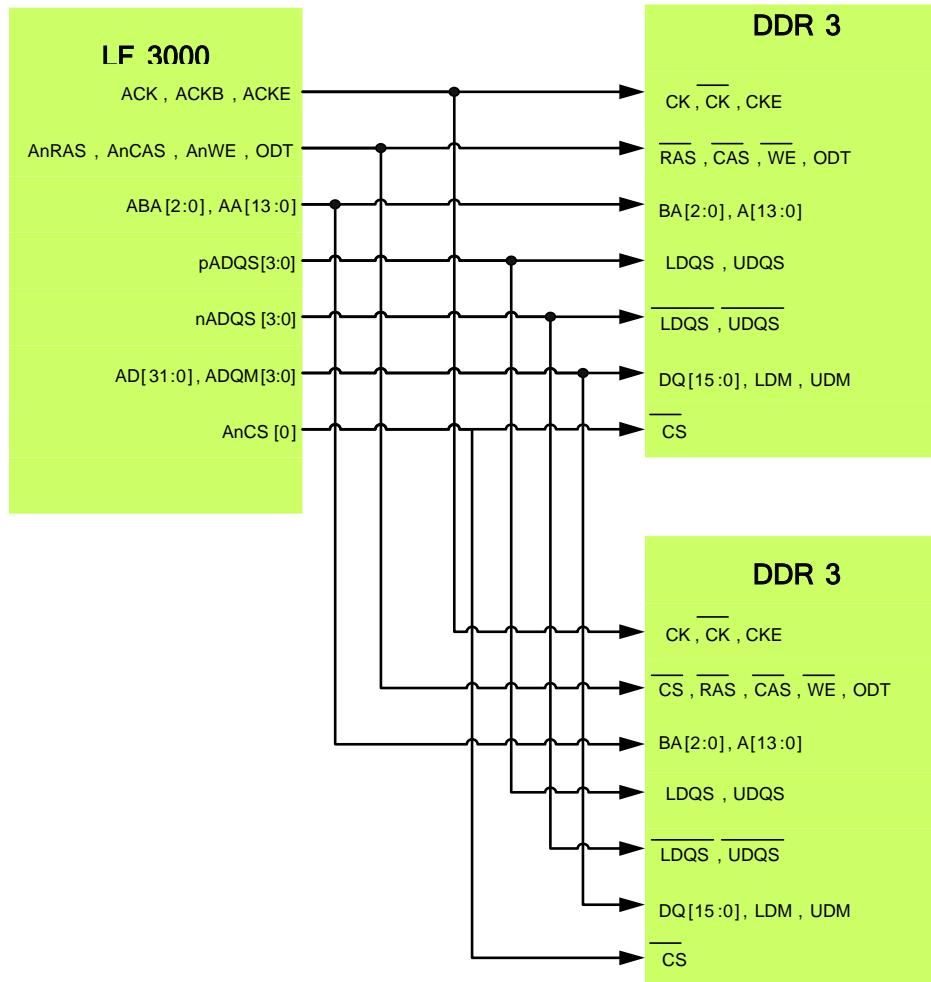


Figure 14-7. 16bit Data bus width SDRAM Interface

14.2.8 NAND Overview

14.2.8.1 Normal Access Sequence

Read Cycle

- Write NAND Flash Command at **NFCMD** Register.
- Write Address to NFADDR Register with respect to NAND Flash Address Type. (Refer to NAND Flash data book)
- Check **IRQPEND** bit.
- Read ECC Data from NAND Flash Spare Array and Write **ORGECC** Register.
- Read Data (512/1024byte) from NAND Flash Main Array.
- Check **NFDecECCDone** Register

- Check **NFCHECKERROR** Register

Write Cycle

- Write NAND Flash Command to **NFCMD** Register. (Refer to NAND Flash data book)
- Access address of memory which tries to access to **NFADDR** Register by 3~5 time according to the sort of NAND Flash. (Refer to NAND Flash data book)
- Write data (512/1024byte) through **NFDATA** Register.
- Check **NFECCDONE** Register.
- Read the result of ECC through **NFECC** Register. (Small block only)
- Write NAND Flash Command to **NFCMD** Register. (Refer to NAND Flash data book)
- Check whether NAND Flash is ready or not by reading **NFCONTROL.IRQPEND** bit. (For the exact sequence of NAND Flash, Please refer to NAND Flash data book)

14.2.8.2 ECC (BCH)

Feature

- Hardware ECC generation, detection and indication (software correction)
4/8/16/24/40/60 bit error correction and detection
- Error Detection Code/Error Correction Code (EDC/ECC)
NAND-based MLC flash has error weakness therefore error handling method is required. NXP3200 can perform EDC (error detecting codes) and ECC (error correction codes) and both based on BCH (Bose-Chadhuri-Hocquenghem) algorithm. EDC is run by hardware to reduce CPU overload and increase the CPU performance. In contrast, ECC is run by software. Parity bit is calculated on every 512/1024 byte page. Syndrome is calculated when each data is read from NAND flash and the value is used in error correction operation.
- Hardware ECC generation reset
 - This reset is asserted when NAND address or command register is written by any value.
 - This reset is also asserted when ECCRST bit (NFCONTROL register [11] bit) register is set to '1'
 - This reset initializes NFECCCL, NFECCCH, NFCNT, NFECCSTATUS, NFSYNDRONE0~7 Registers.

14.3 Register Summary

Symbol	Description
DREX	0xC00E0000
DDDRPHY	0xC00E1000
MCUS_ADDR	0xC0051000

14.3.1 DREX Register

Bit	R/W	Symbol	Description	Reset Value
Controller Control Register (ConControl)				
Address : 0xC00E_0000				
[31:29]		RESERVED	Should be zero	0x0
[28]	R/W	DFI_INIT_START	DFI PHY initialization start This field is used to initialize DFI PHY. Set this field to 1 to initialize DFI PHY and set this field to 0 after received dfi_init_complete of PhyStatus register.	0x0
[27:16]	R/W	TIMEOUT_LEVEL0	Default Timeout Cycles This counter prevents transactions in the AXI request FIFO from starvation. This counter starts if a new AXI transaction comes into the request FIFO. If the counter becomes zero, the corresponding FIFO has the highest priority during BRB arbitration. Refer to Section 1.7 . Quality of Service for detailed information	0xFFFF
[15]		RESERVED	Should be zero	0x0
[14:12]	R/W	RD_FETCH	Read Data Fetch Cycles 0xn = n MBCLK cycles (MBCLK : DREX-1 core clock) The recommended value of this field is 0x2 for LPDDR3 800MHz memory clock and other cases are 0x1. This register is for the unpredictable latency of read data coming from memory devices by tDQSCK variation or the board flying time. The read fetch delay of PHY read FIFO must be controlled by this parameter. The controller will fetch read data from PHY after read_latency + n MBCLK cycles. Refer to Section 1.3 for detailed information.	0x1
[11:9]		RESERVED	Should be zero	0x0
[8]	R	EMPTY	Empty Status 0x0 = Not Empty, 0x1 = Empty There is no AXI transaction in memory controller.	0x1
[7:6]		RESERVED	Should be zero	0x0
[5]	R/W	AREF_EN	Auto Refresh Counter 0x0 = Disable, 0x1 = Enable Enable this to decrease the auto refresh counter by 1 at the rising edge of the rclk	0x0
[4]		RESERVED	Should be zero	0x0
[3]	R/W	IO_PD_CON	I/O Powerdown Control in Low Power Mode(through LPI) 0x0 = Use programmed ctrl_pd and pulldown control 0x1 = Automatic control for ctrl_pd and pulldown control If this value is set to 0x1 and in low power mode through LPI, DREX-1 automatically sets powerdown enable for input buffer of I/O and pulldown disable for dq and dqs in powerdown mode	0x0

Bit	R/W	Symbol	Description	Reset Value
			If this value is set to 0x0, DREX-1 only sends programmed ctrl_pd value and pulldown control.	
[2:1]	R/W	CLK_RATIO	Clock Ratio of Bus Clock to Memory Clock 0x0 = freq.(MBCLK): freq.(MBCLK) = 1: 1, 0x1 ~ 0x3 = Reserved	0x0
[0]		RESERVED	Should be zero	0x0
Memory Control Register (MemControl)				
Address : 0xC00E_0004				
[31:27]		RESERVED	Should be zero	0x0
[26:25]	R/W	MRR_BYTE	Mode Register Read byte lane location 0x0 = memory dq[7:0] 0x1 = memory dq[15:8] 0x2 = memory dq[23:16] 0x3 = memory dq[31:24]	0x0
[24]	R/W	PZQ_EN	DDR3 periodic ZQ(ZQCS) enable Note that after exit from self refresh, ZQ function is required by the software. The controller does not support ZQ calibration command to be issued in parallel to DLL lock time when coming out of self refresh. Turn-on only when using DDR3. The periodic ZQ interval is defined by t_pzq in TIMINGPZQ register.	0x0
[23]		RESERVED	Should be zero	0x0
[22:20]	R/W	BL	Memory Burst Length 0x0~1 = Reserved, 0x2 = 4, 0x3 = 8, 0x4 ~ 0x7 = Reserved In case of LPDDR2-S4, the controller only supports burst length 4. In case of DDR3 and LPDDR3, the controller only supports burst length 8.	0x2
[19:16]	R/W	NUM_CHIP	Number of Memory Chips 0x0 = 1 chip, 0x1 = 2 chips, 0x2 ~ 0xf = Reserved	0x0
[15:12]	R/W	MEM_WIDTH	Width of Memory Data Bus 0x0 ~ 0x1 = Reserved, 0x2 = 32-bit, 0x3 ~ 0xf= Reserved	0x2
[11:8]	R/W	MEM_TYPE	Type of Memory 0x0 ~ 0x4 = Reserved, 0x5 = LPDDR2-S4, 0x6 = DDR3, 0x7 = LPDDR3, 0x8 ~ 0xf = Reserved	0x6
[7:6]	R/W	ADD_LAT_PALL	Additional Latency for PALL in MBCLK cycle 0x0 = 0 cycle, 0x1 = 1 cycle 0x2 = 2 cycle, 0x3 = Reserved If all banks precharge command is issued, the latency of pre-charging will be tRP + add_lat_pall	0x0
[5]	R/W	DSREF_EN	Dynamic Self Refresh	0x0

Bit	R/W	Symbol	Description	Reset Value
			0x0 = Disable, 0x1 = Enable Refer to Section 1.5.2 . Dynamic power down for detailed information. In DDR3, this feature is not supported. This feature must be turn-off when using DDR3.	
[4]	R/W	TP_EN	Timeout Precharge 0x0 = Disable, 0x1 = Enable If tp_en is enabled, it automatically precharges an open bank after a specified amount of mclk cycles (if no access has been made in between the cycles) in an open page policy. If Prefetch-Config.tp_crt bit-field is set, it specifies the amount of mclk cycles to wait until timeout precharge precharges the open bank. Refer to Section 1.6.2 . Timeout precharge for detailed information.	0x0
[3:2]	R/W	DPWRDN_TYPE	Type of Dynamic Power Down 0x0 = Active/precharge power down, 0x1 = Forced precharge power down 0x2 ~ 0x3 = Reserved Refer to Section 1.5.2 . Dynamic power down for detailed information.	0x0
[1]	R/W	DPWRDN_EN	Dynamic Power Down 0x0 = Disable, 0x1 = Enable	0x0
[0]	R/W	CLK_STOP_EN	Dynamic Clock Control 0x0 = Always running, 0x1 = Stops during idle periods This feature is only supported with LPDDR2/LPDDR3. Refer to Section 1.5.4 . Clock stop for detailed information.	0x1
Memory Chip0 Configuration Register (MemConfig0)				
Address : 0xC00E_0008				
[31:16]		RESERVED	Should be zero	0x0
[15:12]	R/W	CHIP_MAP	Address Mapping Method (AXI to Memory) 0x0 = Linear ({bank, row, column, width}), 0x1 = Interleaved ({row, bank, column, width}), 0x2 ~ 0xf = Reserved	0x1
[11:8]	R/W	CHIP_COL	Number of Column Address Bits 0x0 = Reserved, 0x1 = Reserved, 0x2 = 9 bits, 0x3 = 10 bits, 0x4 = Reserved, 0x5 ~ 0xf = Reserved	0x3
[7:4]	R/W	CHIP_ROW	Number of Row Address Bits 0x0 = 12 bits, 0x1 = 13 bits, 0x2 = 14 bits, 0x3 = 15 bits, 0x4 = 16 bits, 0x5 ~ 0xf = Reserved	0x1
[3:0]	R/W	CHIP_BANK	Number of Banks 0x0 = Reserved,	0x2

Bit	R/W	Symbol	Description	Reset Value
			0x1 = Reserved, 0x2 = 4 banks, 0x3 = 8 banks, 0x4 ~ 0xf = Reserved	
Memory Chip1 Configuration Register (MemConfig1)				
Address : 0xC00E_000C				
[31:16]		RESERVED	Should be zero	0x0
[15:12]	R/W	CHIP_MAP	Address Mapping Method (AXI to Memory) 0x0 = Linear ({bank, row, column, width}), 0x1 = Interleaved ({row, bank, column, width}), 0x2 ~ 0xf = Reserved	0x1
[11:8]	R/W	CHIP_COL	Number of Column Address Bits 0x0 = Reserved, 0x1 = Reserved, 0x2 = 9 bits, 0x3 = 10 bits, 0x4 = Reserved, 0x5 ~ 0xf = Reserved	0x3
[7:4]	R/W	CHIP_ROW	Number of Row Address Bits 0x0 = 12 bits, 0x1 = 13 bits, 0x2 = 14 bits, 0x3 = 15 bits, 0x4 = 16 bits, 0x5 ~ 0xf = Reserved	0x1
[3:0]	R/W	CHIP_BANK	Number of Banks 0x0 = Reserved, 0x1 = Reserved, 0x2 = 4 banks, 0x3 = 8 banks, 0x4 ~ 0xf = Reserved	0x2
Memory Direct Command Register (DirectCmd)				
Address : 0xC00E_0010				
[31:28]		RESERVED	Should be zero.	0x0
[27:24]	R/W	CMD_TYPE	Type of Direct Command 0x0 = MRS/EMRS (mode register setting), 0x1 = PALL (all banks precharge), 0x2 = PRE (per bank precharge), 0x3 = DPD (deep power down), 0x4 = REFS (self refresh), 0x5 = REFA (auto refresh), 0x6 = CKEL (active/precharge power down), 0x7 = NOP (exit from active/precharge power down or deep power down), 0x8 = REFSX (exit from self refresh), 0x9 = MRR (mode register reading), 0xa = ZQINIT(ZQ calibration init.) 0xb = ZQOPER(ZQ calibration long)	0x0

Bit	R/W	Symbol	Description	Reset Value
			<p>0xc = ZQCS(ZQ calibration short)</p> <p>0xd ~ 0xf = Reserved</p> <p>When a direct command is issued, AXI masters must not access memory. It is strongly recommended to check the command queue "state by CmdChip0/1_empty and the chip FSM in the ChipStatus register before issuing a direct command. The chip status must be checked before issuing a direct command.</p> <p>And clk_stop_en, dynamic power down, dynamic self refresh, force precharge function (MemControl register) and sl_dll_dyn_con (PhyControl0 register) must be disabled.</p> <p>MRS/EMRS or MRR commands should be issued if all banks are in idle state.</p> <p>If MRS/EMRS or MRR is issued to LPDDR2-S4/LPDDR3, the CA pins must be mapped as follows.</p> <p>MA[7:0] = {cmd_addr[1:0], cmd_bank[2:0], cmd_addr[12:10]}, OP[7:0] = cmd_addr[9:2]</p> <p>In DDR3, self refresh related timing such as tCKESR/tCKSRE/tCKSRX should be check by software.</p> <p>Note that do not write reserved value to this field.</p>	
[23:21]		RESERVED	Should be zero.	0x0
[20]	R/W	CMD_CHIP	<p>Chip Number to send the direct command to</p> <p>0 = Chip 0 1 = Chip 1</p>	0x0
[19]		RESERVED	Should be zero.	0x0
[18:16]	R/W	CMD_BANK	<p>Related Bank Address when issuing a direct command</p> <p>To send a direct command to a chip, additional information such as the bank address is required. This register is used in such situations</p>	0x0
[15:0]	R/W	CMD_ADDR	<p>Related Address value when issuing a direct command</p> <p>To send a direct command to a chip, additional information such as the address is required. This register is used in such situations.</p>	0x0
Precharge Policy Configuration Register (PrechConfig)				
Address : 0xC00E_0014				
[31:24]	R/W	TP_CNT	<p>Timeout Precharge Cycles</p> <p>0xn = n MBCLK cycles, The minimum value of this field is 0x2</p> <p>If the timeout precharge function (MemControl.tp_en) is enabled and the timeout precharge counter becomes zero, the controller forces the activated memory bank into the pre-charged state. Refer to Section 1.6.2 .Timeout precharge for detailed information.</p>	0xFF
[23:16]		RESERVED	Should be zero	0x0
[15:8]	R/W	CHIP1_POLICY	<p>Memory Chip1 Precharge Bank Selective Policy</p> <p>0x0 = Open page policy, 0x1 = Close page (auto precharge) policy</p> <p>chip1_policy[n], n is the bank number of chip1.</p> <p>Open Page Policy: After a READ or WRITE, the row that was accessed is left open.</p> <p>Close Page (Auto Precharge) Policy: Right after a READ or WRITE command, memory devices automatically precharges the bank.</p> <p>This is a bank selective precharge policy. For example, if chip1_policy[2] is 0x0, bank2 of chip1 has an open page policy and if chip1_policy[6] is 0x1, bank6 of chip1 has a close page policy. Refer to Section 1.6.1 . Bank Selective Precharge for detailed information.</p>	0x0
[7:0]	R/W	CHIP0_POLICY	<p>Memory Chip0 Precharge Bank Selective Policy</p> <p>0x0 = Open page policy, 0x1 = Close page (auto precharge) policy</p> <p>Chip0_policy[n], n is the bank number of chip0.</p> <p>This is for memory chip0.</p>	0x0

Bit	R/W	Symbol	Description	Reset Value
PHY Control0 Register (PhyControl0)				
Address : 0xC00E_0018				
[31]	R/W	MEM_TERM_EN	Termination Enable for Memory At high-speed memory operation, it can be necessary to turn on termination resistance in memory devices. This register controls an ODT pin of a memory device.	0x0
[30]	R/W	PHY_TERM_EN	Termination Enable for PHY At high-speed memory operation, it can be necessary to turn on termination resistance in the PHY.	0x0
[29]		CTRL_SHGATE	Duration of DQS Gating Signal This field controls the gate control signal In LPDDR2-S4/LPDDR3, this field should be 1 "b0" of clock frequency. In DDR3, according to memory clock, set the value like below. 1'b0 = (gate signal length = "burst length / 2" (<= 200MHz)) 1'b1 = (gate signal length = "burst length / 2" - 1 (> 200MHz))	0x0
[28:24]	R/W	CTRL_PD	Input Gate for Power Down If this field is set, input buffer is off for power down. This field should be 0 for normal operation. Ctrl_pd[3:0] = for each data slice, Ctrl_pd[4] = for control slice.	0x0
[23:7]		RESERVED	Should be zero	0x0
[6:4]	R/W	DQS_DELAY	Delay Cycles for DQS Cleaning This register is to enable PHY to clean incoming DQS signals delayed by external circumstances. If DQS is coming with read latency plus n memory clock cycles, this register must be set to n memory clock cycles.	0x0
[3]	R/W	FP_RESET	Force DLL Resynchronization	0x0
[2]		RESERVED	Should be zero	0x0
[1]	R/W	SL_DLL_DYN_CON	Turn On PHY Slave DLL Dynamically 0 = Disable 1 = Enable	0x0
[0]	R/W	MEM_TERM_CHIPS	Memory termination between chips 0 = Disable 1 = Enable This field is only valid when num_chip is 0x1(2 chips) in MemControl register and DDR3.	0x0
RESERVED				
Address : 0xC00E_001C ~ 0xC00E_0024				
Dynamic Power Down Configuration Register (PwrDnConfig)				
Address : 0xC00E_0028				
[31:16]	R/W	DSREF_CYC	Number of Cycles for dynamic self refresh entry 0xn = n MBCLK cycles, The minimum value of this field is 0x2. If the command queue is empty for n+1 cycles, the controller forces memory devices into self refresh state. Refer to Section 1.5.3 . Dynamic self refresh for detailed information.	0xFFFF
[15:8]		RESERVED	Should be zero	0x0
[7:0]	R/W	DPWRDN_CYC	Number of Cycles for dynamic power down entry 0xn = n MBCLK cycles, The minimum value of this field is 0x2. If the command queue is empty for n+1 cycles, the controller forces the memory device into active/precharge power down state. Refer to Section 1.5.2 . Dynamic power down for detailed information.	0xFF

Bit	R/W	Symbol	Description	Reset Value
AC Timing Register for Periodic ZQ(ZQCS) of Memory (TimingPZQ)				
Address : 0xC00E_002C				
[31:24]		RESERVED	Should be zero	0x0
[23:0]	R/W	T_PZQ	<p>Average Periodic ZQ Interval(Only in DDR3)</p> <p>$t_{refi}(t_{refi} * T(rclk)) * t_{pzq}$ should be less than or equal to the minimum value of memory periodic ZQ interval,</p> <p>for example, if rclk frequency is 12MHz, t_{refi} is set to 93 and ZQ interval is 128ms then the following value should be programmed into it: $128\text{ ms} * 12\text{ MHz} / 93 = 16516$</p> <p>The minimum value is 2.</p>	0x4084
AC Timing Register for Auto Refresh of Memory (TimingAref)				
Address : 0xC00E_0030				
[31:16]		RESERVED	Should be zero	0x0
[15:0]	R/W	T_REFI	<p>Average Periodic Refresh Interval</p> <p>$t_{refi} * T(rclk)$ should be less than or equal to the minimum value of memory tREFI (all bank),</p> <p>for example, for the all bank refresh period of 7.8us, and an rclk frequency of 12MHz, the following value should be programmed into it: $7.8\text{ us} * 12\text{ MHz} = 93$</p>	0x5D
AC Timing Register for the Row of Memory (TimingRow)				
Address : 0xC00E_0034				
[31:24]	R/W	T_RFC	<p>Auto refresh to Active / Auto refresh command period, in MBCLK cycles</p> <p>$t_{rfc} * T(MBCLK)$ should be greater than or equal to the minimum value of memory tRFC and the minimum value is 17 if PHY is running with dll on. In FPGA with low frequency and dll is off, the minimum value is 3.</p>	0x1F
[23:20]	R/W	T_RRD	<p>Active bank A to Active bank B delay, in MBCLK cycles</p> <p>$t_{rrd} * T(MBCLK)$ should be greater than or equal to the minimum value of memory tRRD.</p> <p>The minimum value is 2.</p>	0x2
[19:16]	R/W	T_RP	<p>Precharge command period, in MBCLK cycles</p> <p>$t_{rp} * T(MBCLK)$ should be greater than or equal to the minimum value of memory tRP.</p> <p>The minimum value is 2.</p>	0x3
[15:12]	R/W	T_RCD	<p>Active to Read or Write delay, in MBCLK cycles $t_{rcd} * T(MBCLK)$ should be greater than or equal to the minimum value of memory tRCD + T(MBCLK)/2.</p> <p>For example,</p> <p>tRCD in memory specification is 13.75ns and MBCLK is 3.0ns,</p> <p>$t_{rcd} * 3\text{ns} \geq 13.75\text{ns} + 1.5\text{ns}$</p> <p>The right value for t_{rcd} is 6.</p> <p>The minimum value is 2.</p>	0x3
[11:6]	R/W	T_RC	<p>Active to Active period, in MBCLK cycles</p> <p>$t_{rc} * T(MBCLK)$ should be greater than or equal to the minimum value of memory tRC.</p> <p>The minimum value is 2.</p>	0xA
[5:0]	R/W	T_RAS	<p>Active to Precharge command period, in MBCLK cycles</p> <p>$t_{ras} * T(MBCLK)$ should be greater than or equal to the minimum value of memory tRAS + T(MBCLK)/2.</p> <p>For example,</p> <p>tRAS in memory specification is 35ns and MBCLK is 3.0ns.</p> <p>$t_{ras} * 3\text{ns} \geq 35\text{ns} + 1.5\text{ns}$</p> <p>The right value for t_{ras} is 13.</p> <p>The minimum value is 2.</p>	0x6
AC Timing Register for the Data of Memory (TimingData)				

Bit	R/W	Symbol	Description	Reset Value
Address : 0xC00E_0038				
[31:28]	R/W	T_WTR	Internal write to Read command delay, in MBCLK cycles t_wtr * T(MBCLK) should be greater than or equal to the minimum value of memory tWTR. In LPDDR2-S4/LPDDR3 t_wtr is max(2tCK, tWTR) and in DDR3 t_wtr is max(4tCK, tWTR). And then this value should be changed in MBCLK cycles. The minimum value is 2.	0x1
[27:24]	R/W	T_WR	Write recovery time, in MBCLK cycles t_wr * T(MBCLK) should be greater than or equal to the minimum value of memory tWR. The minimum value is 2.	0x2
[23:20]	R/W	T RTP	Internal read to Precharge command delay, in MBCLK cycles t_rtp * T(MBCLK) should be greater than or equal to the minimum value of memory tRTP. The minimum value is 2.	0x3
[19:17]	R/W	RESERVED	Should be zero	0x0
[16]	R/W	T_W2W_C2C	Additional Write to Write delay in chip to chip case in MBCLK cycles. If mem_term_en of PhyControl0 register(offset addr = 0x18) is enabled, then this value will be set to 1 automatically.	0x0
[15]	R/W	T_R2R_C2C	Additional Read to Read delay in chip to chip case in MBCLK cycles. If mem_term_chips of PhyControl0 register(offset addr = 0x18) is enabled, then this value will be set to 1 automatically.	0x0
[14:12]		DQSCK	tDQSCK in memory clock cycles In DDR3, this field should set to 0. In LPDDR2/3, this field should be set to RU(tDQSCK max/tCK). tDQSCK max is 5.5ns in LPDDR2/3.	0x3
[11:8]	R/W	WL	Write data latency (for LPDDR2-S4/LPDDR3/DDR3), in memory clock cycles wl should be greater than or equal to the minimum value of memory WL. There is no restriction with LPDDR2-S4 but there is restriction with LPDDR3 and DDR3 like below. In LPDDR3, the minimum wl is 5 and in DDR3, the minimum wl is 6.	0x6
[7:4]		RESERVED	Should be zero	0x0
[3:0]	R/W	RL	Read data latency (for LPDDR2-S4/LPDDR3/DDR3), in memory clock cycles rl should be greater than or equal to the minimum value of memory RL	0xC
AC Timing Register for the Power modes of Memory (TimingPower)				
Address : 0xC00E_003C				
[31:26]	R/W	T_FAW	Four Active Window(for LPDDR2-S4/LPDDR3/DDR3) t_faw * T(MBCLK) should be greater than or equal to the minimum value of memory tFAW. The minimum value is 2.	0xE
[25:16]	R/W	T_XSR	Self refresh exit power down to next valid command delay, in cycles t_xsr * T(MBCLK) should be greater than or equal to the minimum value of memory tXSR. In DDR3, this value should be set to tXSDLL. The minimum value is 2.	0x1B
[15:8]	R/W	T_XP	Exit power down to next valid command delay, in cycles t_xp * T(MBCLK) should be greater than or equal to the minimum value of memory tXP In DDR3, this value should be set to tXPDLL. In DDR3 even though "fast exit" is programmed in MRS, tXPDLL is applied. In DDR3, tXPDLL is likely max(10nCK,24ns). So note that t_xp should set to max(5nMBCLK,24ns/MBCLK period). The minimum value is 2.	0x4

Bit	R/W	Symbol	Description	Reset Value
[7:4]	R/W	T_CKE	CKE minimum pulse width (minimum power down mode duration), in cycles t_cke should be greater than or equal to the minimum value of memory tCKE. The minimum value is 2.	0x2
[3:0]	R/W	T_MRД	Mode Register Set command period, in cycles t_mrd should be greater than or equal to the minimum value of memory tMRД In DDR3, this parameter should be set to tMOD value. The minimum value is 2.	0x2
PHY Status Register (PhyStatus)				
Address : 0xC00E_0040				
[31:15]	R	RESERVED	Should be zero	0x0
[14]	R	RDLVL_COMPLETE	Read Level Completion	0x0
[13:4]	R	RESERVED	Should be zero	0x0
[3]	R	DFI_INIT_COMPLETE	DFI PHY initialization complete 0 = Initialization has not been finished 1 = Initialization has been finished	0x0
[2:0]	R	RESERVED	Should be zero	0x0
RESERVED				
Address : 0xC00E_0044				
Memory ChipStatus Register (ChipStatus)				
Address : 0xC00E_0048				
[31:16]		RESERVED	Should be zero	0x0
[15:12]	R	CHIP_DPD_STATE	Chip is in the deep powerdown state	0x0
[11:8]	R	CHIP_SREF_STATE	Chip is in the self-refresh state	0x0
[7:4]	R	CHIP_PD_STATE	Chip is in the powerdown state	0x0
[3:0]	R	CHIP_BUSY_STATE	Chip is in the busy state [0] = chip0 busy state [1] = chip1 busy state [2] = chip2 busy state [3] = chip3 busy state	0x0
RESERVED				
Address : 0xC00E_004C ~ 0xC00E_0050				
Memory Mode Registers Status Register (MrStatus)				
Address : 0xC00E_0054				
[31:8]	R	RESERVED	Should be zero	0x0
[7:0]	R	MR_STATUS	Mode Registers Status	0x0
RESERVED				
Address : 0xC00E_0058 ~ 0xC00E_005C				
Quality of Service Control Register n (QosControl n)				
Address : 0xC00E_0060 ~ 0xC00E_00D8				
[31:12]	R/W	RESERVED	Should be zero	0x0
[11:0]	R/W	CFG_QOS	QoS Cycles 0xn = n MBCLK cycles The matched ARID uses this value for its timeout counters instead of ConControl.timeout_cnt.	0xFF

Bit	R/W	Symbol	Description	Reset Value
<NOTE> QosControl n : n=0~15				
QosControl0 : Address = DREX + 0x0060				
QosControl1 : Address = DREX + 0x0068				
QosControl2 : Address = DREX + 0x0070				
QosControl3 : Address = DREX + 0x0078				
QosControl4 : Address = DREX + 0x0080				
QosControl5 : Address = DREX + 0x0088				
QosControl6 : Address = DREX + 0x0090				
QosControl7 : Address = DREX + 0x0098				
QosControl8 : Address = DREX + 0x00A0				
QosControl9 : Address = DREX + 0x00A8				
QosControl10 : Address = DREX + 0x00B0				
QosControl11 : Address = DREX + 0x00B8				
QosControl12 : Address = DREX + 0x00C0				
QosControl13 : Address = DREX + 0x00C8				
QosControl14 : Address = DREX + 0x00D0				
QosControl15 : Address = DREX + 0x00D8				
RESERVED				
Address : 0xC00E_00DC ~ 0xC00E_00F0				
Write Training Configuration Register (WrTraConfig)				
Address : 0xC00E_00F4				
[31:16]	R/W	ROW_ADDR	Row Address for Write Training	0x0
[15:4]		RESERVED	Should be zero	0x0
[3:1]	R/W	BANK	Bank Address for Write Training	0x0
[0]	R/W	WRITE_TRAINING_EN	Write Training Enable Use this field to issue ACT command. Before setting this field, below things should be finished. Please see PHY manual. - Set write latency before write training(PHY's control register 26) - Set write training mode(PHY's control register 2) 0x0 = Disable, 0x1 = Enable (Issue ACT command)	0x0
Read Leveling Configuration Register (RdlvConfig)				
Address : 0xC00E_00F8				
[31:2]		RESERVED	Should be zero	0x0
[1]	R/W	CTRL_RDLVL_DATAB_EN	Data eye training enable	0x0
[0]	R/W	CTRL_RDLVL_GATE_EN	Gate training enable This is only valid for DDR3 case. If LPDDR2-S4/LPDDR3 is used, this field must be set to 0x0. When ctrl_rdlv_en = 1, Read leveling offset values will be used instead of ctrl_offset*. If read leveling is used, this value should be high during operation. This field should be set after dfi_init_complete is asserted. 0x0 = Disable, 0x1 = Enable	0x0
RESERVED				
Address : 0xC00E_00FC				
BRB Reservation Control Register (BRBRSVCONTROL)				

Bit	R/W	Symbol	Description	Reset Value
Address : 0xC00E_0100				
[31:8]		RESERVED	Should be zero	0x0
[7]	R/W	BRB_RSV_EN_W3	Enable Write-BRB reservation for AXI port 3	0x0
[6]	R/W	BRB_RSV_EN_W2	Enable Write-BRB reservation for AXI port 2	0x0
[5]	R/W	BRB_RSV_EN_W1	Enable Write-BRB reservation for AXI port 1	0x0
[4]	R/W	BRB_RSV_EN_W0	Enable Write-BRB reservation for AXI port 0	0x0
[3]	R/W	BRB_RSV_EN_R3	Enable Read-BRB reservation for AXI port 3	0x0
[2]	R/W	BRB_RSV_EN_R2	Enable Read-BRB reservation for AXI port 2	0x0
[1]	R/W	BRB_RSV_EN_R1	Enable Read-BRB reservation for AXI port 1	0x0
[0]	R/W	BRB_RSV_EN_R0	Enable Read-BRB reservation for AXI port 0	0x0
BRB Reservation Configuration Register (BRBRSVCONFIG)				
Address : 0xC00E_0104				
[31:28]	R/W	BRB_RSV_TH_W3	Write-BRB reservation threshold for AXI port 3 Write request from AXI port3 does not serviced when Write-BRB reservation is enabled for AXI port 3 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w3.	0x8
[27:24]	R/W	BRB_RSV_TH_W2	Enable Write-BRB reservation for AXI port 2 Write request from AXI port2 does not serviced when Write-BRB reservation is enabled for AXI port 2 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w2.	0x8
[23:20]	R/W	BRB_RSV_TH_W1	Enable Write-BRB reservation for AXI port 1 Write request from AXI port1 does not serviced when Write-BRB reservation is enabled for AXI port 1 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w1.	0x8
[19:16]	R/W	BRB_RSV_TH_W0	Enable Write-BRB reservation for AXI port 0 Write request from AXI port0 does not serviced when Write-BRB reservation is enabled for AXI port 0 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w0.	0x8
[15:12]	R/W	BRB_RSV_TH_R3	Enable Read-BRB reservation for AXI port 3 Read request from AXI port3 does not serviced when Read-BRB reservation is enabled for AXI port 3 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w3.	0x8
[11:8]	R/W	BRB_RSV_TH_R2	Enable Read-BRB reservation for AXI port 2 Read request from AXI port2 does not serviced when Read-BRB reservation is enabled for AXI port 2 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w2.	0x8
[7:4]	R/W	BRB_RSV_TH_R1	Enable Read-BRB reservation for AXI port 1 Read request from AXI port1 does not serviced when Read-BRB reservation is enabled for AXI port 1 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w1.	0x8
[3:0]	R/W	BRB_RSV_TH_R0	Enable Read-BRB reservation for AXI port 0 Read request from AXI port0 does not serviced when Read-BRB reservation is enabled for AXI port 0 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w0.	0x8
BRB QoS Configuration Register (BRBQOSCONFIG)				
Address : 0xC00E_0108				
[31:12]	R	RESERVED	Should be zero	0x0
[11:0]	R/W	BRB_QOS_TIMER_DEC	BRB timer decrementing size for QoS. The timer for request in BRB decreases by brb_qos_timer_dec for the following cases. - When the BRB is full - When the request is from the AXI port whose data buffer is full.	0x10

Bit	R/W	Symbol	Description	Reset Value
Memory Chip0 Base Configuration Register (MemBaseConfig0)				
Address : 0xC00E_010C				
[31:27]		RESERVED	Should be zero	0x0
[26:16]	R/W	CHIP_BASE	AXI Base Address AXI base address [34:24] = chip_base, For example, if chip_base = 0x20, then AXI base address of memory chip0 becomes 0x2000_0000.	0x20
[15:11]		RESERVED	Should be zero	0x0
[10:0]	R/W	CHIP_MASK	AXI Base Address Mask Upper address bit mask to determine AXI offset address of memory chip0. 0 = Corresponding address bit is not to be used for comparison 1 = Corresponding address bit is to be used for comparison For example, if chip_mask = 0x7F8, then AXI offset address becomes 0x0000_0000 ~ 0x07FF_FFFF. If AXI base address of memory chip0 is 0x2000_0000, then memory chip0 has an address range of 0x2000_0000 ~ 0x27FF_FFFF.	0x7F8
<Note> DRAM Start Address : 0x0000_0000 DRAM End Address : 0x7FFF_FFFF				
Memory Chip1 Base Configuration Register (MemBaseConfig1)				
Address : 0xC00E_0110				
[31:27]		RESERVED	Should be zero	0x0
[26:16]	R/W	CHIP_BASE	AXI Base Address AXI base address [34:24] = chip_base, For example, if chip_base = 0x20, then AXI base address of memory chip0 becomes 0x2000_0000.	0x20
[15:11]		RESERVED	Should be zero	0x0
[10:0]	R/W	CHIP_MASK	AXI Base Address Mask Upper address bit mask to determine AXI offset address of memory chip0. 0 = Corresponding address bit is not to be used for comparison 1 = Corresponding address bit is to be used for comparison For example, if chip_mask = 0x7F8, then AXI offset address becomes 0x0000_0000 ~ 0x07FF_FFFF. If AXI base address of memory chip0 is 0x2000_0000, then memory chip0 has an address range of 0x2000_0000 ~ 0x27FF_FFFF.	0x7F8
RESERVED				
Address : 0xC00E_0114 ~ 0xC00E_011C				
Write Leveling Configuration Register0 (WRLVLCONFIG0)				
Address : 0XC00E_0120				
[31:8]		RESERVED	Should be zero	0x0
[7:4]	R/W	T_WLO	Write Leveling Output Delay t_wlo * T(PCLK) should be greater than or equal to the minimum value of memory tWLO and the minimum value is 1.	0x1
[3:1]		RESERVED	Should be zero	0x0
[0]	R/W	ODT_ON	Turn On ODT for Write Leveling 0x0 = ODT Turn off 0x1 = ODT Turn on, This field is only for write leveling. Turn on before write leveling and turn off after write leveling is finished. Write leveling procedure is MRS for Write leveling - ODT on - wrvl_wrdata_en - Read ctrl_io_rdata - Change SDLL code of PHY. Refer to PHY manual for details.	0x0

Bit	R/W	Symbol	Description	Reset Value
Write Leveling Configuration Register1 (WRLVLCONFIG1)				
Address : 0XC00E_0124				
[31:1]		RESERVED	Should be zero	0x0
[0]	R/W	WRLVL_WRDATA_EN	Generate dfi_wrdata_en_p0 for Write Leveling Generate 1cycle pulse of dfi_wrdata_en_p0 for write leveling. Write leveling is supported in DDR3 and LPDDR3. Note that if S/W write this field to 1 then, 1 cycle pulse of dfi_wrdata_en_p0 would be generated. Refer to PHY manual for write leveling.	0x0
Write Leveling Status Register (WRLVLSTATUS)				
Address : 0XC00E_0128				
[31:5]	R/W	RESERVED	Should be zero	0x0
[4:0]	R	WRLVL_FSM	Write Leveling Status 5b0_0001: FSM_IDLE 5b0_0010: FSM_SETUP 5b0_0100: FSM_ACCESS 5b0_1000: FSM_DONE 5b1_0000: FSM_TWLO wrM_eq is valid only when wrfl_fsm is FSM_IDLE. Refer to PHY manual for write leveling.	0x0
RESERVED				
Address : 0XC00E_012C ~ 0XC00E_014C				
CTRL_IO_RDATA Register (CTRL_IO_RDATA)				
Address : 0XC00E_0150				
[31:0]	R	CTRL_IO_RDATA	ctrl_io_rdata from PHY	0x0
RESERVED				
Address : 0XC00E_0154 ~ 0XC00E_015C				
CA Calibration Configuration Register0 (CACAL_CONFIG0)				
Address : 0XC00E_0160				
[31:12]	R/W	DFI_ADDRESS_P0	dfi_address_p0 value for CA Calibration This value would be the expected value for comparing the address pattern received from memory.	0xFF
[11:8]	R/W	RESERVED	Should be zero	0x0
[7:4]	R/W	T_ADR	CSN Low to Data Output Delay t_adr * T(PCLK) should be greater than or equal to the min-imun value of memory tADR and the minimum value is 1.	0x1
[3:1]		RESERVED	Should be zero	0x0
[0]	R/W	DEASSERT_CKE	Deassert CKE for CA Calibration 0x0 = Put CKE pin to normal operation 0x1 = Put CKE pin to low This field is only for CA calibration. Deassert CKE before CA calibration and put to normal operation after CA calibration is finished. CA calibration procedure is MRS for CA calibration - Deas-sert CKE - cacal_csn - Read ctrl_io_rdata - Change SDLL code of PHY. Refer to PHY manual for details.	0x0
CA Calibration Configuration Register1 (CACAL_CONFIG1)				
Address : 0XC00E_0164				
[31:1]	R/W	RESERVED	Should be zero	0x0

Bit	R/W	Symbol	Description	Reset Value
[0]	R/W	CACAL_CSN	Generate dfi_csn_p0 for CA Calibration Generate 1cycle pulse of dfi_csn_p0 for CA calibration. CA calibration is supported in LPDDR3. Note that if SW writes this field to 1 then, 1 cycle pulse of dfi_csn_p0 would be generated. Refer to PHY manual for CA calibration.	0x0
CA Calibration Status Register (CACAL_STATUS)				
Address : 0xC00E_0168				
[31:5]	R	RESERVED	Should be zero	0x0
[4:0]	R	CACAL_FSM	Write Leveling Status 5b0_0001: FSM_IDLE 5b0_0010: FSM_SETUP 5b0_0100: FSM_ACCESS 5b0_1000: FSM_DONE 5b1_0000: FSM_TADR CTRL_IO_RDATA are valid only when wrvl_fsm is FSM_IDLE. Refer to PHY manual for CA calibration.	0x0

14.3.2 DDRPHY Register

Bit	R/W	Symbol	Description	Reset Value
PHY_CON0				
Address : 0xC00E_1000				
[31:29]	R	RESERVED	Reserved for future use. This Value has no affect at current system	0
[28:24]	R/W	T_WRWRCMD	It controls the interval between Write and Write during DQ Calibration. This value should be always kept by 5'h17. It will be used for debug purpose	5h17
[23:22]	R/W	RESERVED	Reserved for future use. This Value has no affect at current system	2b00
[21:20]	R/W	CTRL_UPD_RANGE	It decides how many differences between the new lock value and the current lock value which is used in Slave-DLL is needed for updating lock value. 2'b00 : To update always 2'b01 : To ignore the lower 1 bit in ctrl_lock_value 2'b10 : To ignore the lower 2 bits in ctrl_lock_value 2'b11 : To ignore the lower 3 bits in ctrl_lock_value	2b00
[19:17]	R/W	T_WRRDCMD	It controls the interval between Write and Read by cycle unit during Write Calibration. It will be used for debug purpose. 3'b111 : tWTR = 6 cycles 3'b110 : tWTR = 4 cycles	3'b001
[16]	R/W	CTRL_WRLVL_EN (=WRLVL_MODE)	Write Leveling Enable	1'b0
[15]	R/W	RESERVED	Reserved for future use. This Value has no affect at current system	1'b0
[14]	R/W	P0_CMD_EN	1'b0 : Issue Phase1 Read Command during read leveling 1'b1 : Issue Phase0 Read Command during read leveling	1'b0
[13]	R/W	BYTE_RDLVL_EN	Byte Read Leveling enable. It should be set if memory supports toggling only 1 DQ bit except for other 7 bits during read leveling	1'b0
[12:11]	R/W	CTRL_DDR_MODE	2'b00: DDR2 and LPDDR1 2'b01: DDR3 2'b10: LPDDR2 2'b11: LPDDR3	1'b0
[10]	R/W	RESERVED	Reserved for future use. This Value has no affect at current system	1'b0
[9]	R/W	CTRL_DFDQS	1'b0: single-ended DQS 1'b1: differential DQS	1'b1
[8]	R/W	CTRL_SHGATE	This field controls the gate control signal 1'b0: gate signal length = "burst length / 2" + N (DQS Pull-Down mode, ctrl_pulld_dqs[3:0] == 4'b1111, N = 0,1,2,...) 1'b1: gate signal length = "burst length / 2" <input checked="" type="checkbox"/> 1	1'b0
[7]	R/W	CTRL_CKDIS	This field controls the CK/CKB	1'b0

Bit	R/W	Symbol	Description	Reset Value
			1'b0: Clock output is enabled 1'b1: Clock output is disabled	
[6]	R/W	CTRL_ATGATE	If ctrl_atgate=0, Controller should generate ctrl_gate_p*, ctrl_read_p*. If ctrl_atgate=1, PHY will generate ctrl_gate_p*, ctrl_read_p*, but it has some constraints. This setting can be supported only over RL=4, BL and RL should be properly set to operate with ctrl_atgate=1.	1'b1
[5]	R/W	CTRL_READ_DISABLE	Read ODT(On-Die-Termination) Disable Signal. This signal should be one in LPDDR2 or LPDDR3 mode. 1'b1 : drive ctrl_read_p* to 0.(If using LPDDR2 or LPDDR3, ctrl_read_p* will be always 0 by enabling this field.) 1'b0 : drive ctrl_read_p* normally	1'b0
[4]	R/W	CTRL_CMOSRCV	This field controls the input mode of I/O 1'b0: Differential receiver mode for high speed operation 1'b1: CMOS receiver mode for low speed operation (< 200MHz)	1'b0
[3]	R/W	CTRL_READ_WIDTH	The default is 1'b0. We strongly recommend that it should have one more idle cycle between read and write because the variation of data arrival time during read can be greater than 1 cycle. If it is 1'b1, the package and board should be carefully optimized with a short length and it should be used at the low frequency. 1'b0: Termination on period is (BL/2+1.5) cycle (Default) 1'b1: Termination on period is (B/2+1) cycle(Not recommended)	1'b0
[2:0]	R/W	CTRL_FNC_FB	Valid only when {mode_phy, mode_nand, mode_scan, mode_mux} is 4'b0000. For normal operation 3'b000: Normal operation mode. For ATE test purpose 3'b010: External FNC read feedback test mode. 3'b011: Internal FNC read feedback test mode. For Board test purpose 3'b100: External PHY read feedback test mode. When memory is not attached on the board 3'b101: Internal PHY read feedback test mode. mode_highz should be set. 3'b110: Internal PHY write feedback test mode. mode_highz should be set. For Power Down 3'b111: Power Down Mode for SSTL I/O	3'b000
PHY_CON1				
Address : 0xC00E_1004				
[31:28]	R/W	CTRL_GATEADJ	It adjusts the enable time of "ctrl_gate" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	4'h0
[27:24]	R/W	CTRL_READADJ	It adjusts the enable time of "ctrl_read" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	4'h9
[23:20]	R/W	CTRL_GATEDURADJ	It adjusts the duration cycle of "ctrl_gate" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	4'h2
[19:16]	R/W	RDLVL_PASS_ADJ	This field controls how many times "Read" should be operated well to determine if it goes into VWP(Valid Window Period) or not. (default: 4'h1)	4'h1
[15:0]	R/W	RDLVL_RDDATA_ADJ	It decides the pattern to be read during read or write calibration. (default: 16'h0100) 16'h0100 : DDR3 ("byte_rdvl_en=1") 16'h0001 : LPDDR3 ("byte_rdvl_en=1")	16h0100
PHY_CON2				
Address : 0xC00E_1008				
[31:28]	R/W	CTRL_READDURADJ	It adjusts the duration cycle of "ctrl_read" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	0
[27]	R/W	WR DESKEW_EN	DQ Calibration Start Signal to align DQ, DM during write	1'b0
[26]	R/W	WR DESKEW CON	If it is enabled, PHY will use "Write Slave DLL Code" which has got during Read Leveling.	1'b0
[25]	R/W	RDLVL_EN	When rd_cal_mode=1, Read leveling offset values will be used instead of ctrl_offset*. If read leveling is used, this value should be high during operation.	1'b0
[24]	R/W	RDLVL_GATE_EN	When gate_cal_mode=1, Gate leveling offset value will be used instead of ctrl_shifc*. If gate leveling is used, this value should be high during operation	1'b0

Bit	R/W	Symbol	Description	Reset Value
[23]	R/W	RDLVL_CA_EN	When ca_cal_mode=1, CA Calibration offset value will be used and updated	1'b0
[22:16]	R/W	RDLVL_INCR_ADJ	It decides the step value of delay line to increase during read leveling (default: 7'h1, fine step delay). It should be smaller than 7'hf [22:21]=2'b00 : The step value will be "rdlv_incr_adj[20:16]" [22:21]=2'b01 : The step value will be "T/16" [22:21]=2'b10 : The step value will be "T/32" [22:21]=2'b11 : The step value will be "T/64"	7'b1
[15]	R	RESERVED		
[14]	R/W	WRDESKEW_CLEAR	Clear WrDeSkewCode after Write Deskewing	1'b0
[13]	R/W	RDDESKEW_CLEAR	Clear RdDeSkewCode after Read Deskewing	1'b0
[12]	R/W	DLLDESKEWEN	Deskew Code is updated with the latest Master DLL lock value whenever "dfi_ctrlupd_req" is issued from controller during DLLDeskewEn=1. It is required to compensate On-chip VT variation	1'b0
[11:8]	R/W	RDLVL_START_ADJ	It decides the most left-shifted point when read leveling is started and the most right-shifted point when read leveling is ended. [9:8]=2'b00 : The most left-shifted code is 8'h00 [9:8]=2'b01 : The most left-shifted code is T/8 [9:8]=2'b10 : The most left-shifted code is T/8+T/16 [9:8]=2'b11 : The most left-shifted code is T/8-T/16 [11:10]=2'b00 : The most right-shifted Code is 8'hFF [11:10]=2'b01 : The most right-shifted Code is T/2+T/8 [11:10]=2'b10 : The most right-shifted Code is T/2+T/8+T/16 [11:10]=2'b11 : The most right-shifted Code is T/2+T/8-T/16	4'h0
[7]	R/W	RESERVED		
[6]	R/W	INITDESKEWEN	This field should be enabled before DQ Calibration is started	1'b0
[5:2]	R	RESERVED		
[1:0]	R/W	RDLVL_GATEADJ	It determines how much earlier ctrl_gate* is asserted than RDQS when the transition of RDQS is detected. [1:0]=2'b00 : T/2(default) [1:0]=2'b01 : T/4 [1:0]=2'b10 : T/8 [1:0]=2'b11 : T/16	2'b0
PHY_CON3				
Address : 0xC00E_100C				
[31:18]	R	RESERVED	-	-
[17:15]	R/W	CTRL_SHIFTC3	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	3'b010
[14:13]	R	RESERVED	-	-
[12:10]	R/W	CTRL_SHIFTC2	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	3'b010
[9:8]	R	RESERVED	-	-
[7:5]	R/W	CTRL_SHIFTC1	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	3'b010
[4:3]	R	RESERVED	-	-
[2:0]	R/W	CTRL_SHIFTC0	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is	3'b010

Bit	R/W	Symbol	Description	Reset Value
			limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	
PHY_CON4				
Address : 0xC00E_1010				
[31:24]	R/W	CTRL_OFFSETR3	<p>This field can be used to give offset to read DQS.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Read DQS offset amount: $\text{ctrl_offsetr3[7]} = 1$: (tFS: fine step delay) Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$ $\text{ctrl_offsetr3[7]} = 0$: Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$</p>	8'h8
[23:16]	R/W	CTRL_OFFSETR2	<p>This field can be used to give offset to read DQS.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Read DQS offset amount: $\text{ctrl_offsetr3[7]} = 1$: (tFS: fine step delay) Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$ $\text{ctrl_offsetr3[7]} = 0$: Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$</p>	8'h8
[15:8]	R/W	CTRL_OFFSETR1	<p>This field can be used to give offset to read DQS.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Read DQS offset amount: $\text{ctrl_offsetr3[7]} = 1$: (tFS: fine step delay) Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$ $\text{ctrl_offsetr3[7]} = 0$: Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$</p>	8'h8
[7:0]	R/W	CTRL_OFFSETR0	<p>This field can be used to give offset to read DQS.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Read DQS offset amount: $\text{ctrl_offsetr3[7]} = 1$: (tFS: fine step delay) Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$ $\text{ctrl_offsetr3[7]} = 0$: Read DQS 90° delay amount $\square \text{ctrl_offsetr0[6:0] x tFS}$</p>	8'h8
PHY_CON5				
Address : 0xC00E_1014				
[31:8]	R	RESERVED	-	2'h0
[7:0]	R/W	READMODECON	<p>Read Register Mode Control</p> <p>After Gate Leveling, When ReadModeCon[7:0]=hD, Check "Gate Cycle"(0x50) When ReadModeCon[7:0]=hE, Check "Gate Code"(0x50)</p> <p>After Read Calibration, When ReadModeCon[7:0]=h05, Check "VWML"(0x50) When ReadModeCon[7:0]=h06, Check "VWMR"(0x50) When ReadModeCon[7:0]=h07, Check</p>	8'h0

Bit	R/W	Symbol	Description	Reset Value
			<p>"VVMC"(0x50) When ReadModeCon[7:0]=h08, Check "VVMC"(0x50) When ReadModeCon[7:0]=h01, Check "Deskew Code"(0x50)</p> <p>After Write Calibration, When ReadModeCon[7:0]=h05, Check "VWML"(0x50) When ReadModeCon[7:0]=h06, Check "VWMR"(0x50) When ReadModeCon[7:0]=h07, Check "VVMC"(0x50) When ReadModeCon[7:0]=h09, Check "VVMC"(0x50) When ReadModeCon[7:0]=h03, Check "Deskew Code"(0x50)</p> <p>NOTE: The result of VWM* will be over-written after each calibration. Check PHY_CON18(=0x50) according to ReadModeCon[7:0].</p>	
PHY_CON6				
Address : 0xC00E_1018				
[31:24]	R/W	CTRL_OFFSETW3	<p>This field can be used to give offset to write DQ.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Write DQ offset amount :</p> <p>ctrl_offsetw2[7] = 1 : (tFS: fine step delay)</p> <p>Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[6:0] x tFS</p> <p>ctrl_offsetw2[7] = 0 :</p> <p>Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[6:0] x tFS</p>	0x8
[23:16]	R/W	CTRL_OFFSETW2	<p>This field can be used to give offset to write DQ.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Write DQ offset amount :</p> <p>ctrl_offsetw2[7] = 1 : (tFS: fine step delay)</p> <p>Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[6:0] x tFS</p> <p>ctrl_offsetw2[7] = 0 :</p> <p>Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[6:0] x tFS</p>	0x8
[15:8]	R/W	CTRL_OFFSETW1	<p>This field can be used to give offset to write DQ.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Write DQ offset amount :</p> <p>ctrl_offsetw2[7] = 1 : (tFS: fine step delay)</p> <p>Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[6:0] x tFS</p> <p>ctrl_offsetw2[7] = 0 :</p> <p>Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[6:0] x tFS</p>	0x8
[7:0]	R/W	CTRL_OFFSETW0	<p>This field can be used to give offset to write DQ.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Write DQ offset amount :</p> <p>ctrl_offsetw2[7] = 1 : (tFS: fine step delay)</p>	0x8
PHY_CON7				
Address : 0xC00E_101C				
[31:8]	R	RESERVED	Reserved for future use. This Value has no affect at current system.	24'h0
[7:0]	R/W	CTRL_OFFSETW4	<p>This field can be used to give offset to write DQ.</p> <p>If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.</p> <p>Write DQ offset amount :</p>	'0x8

Bit	R/W	Symbol	Description	Reset Value
			ctrl_offsetw2[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[5:0] x tFS ctrl_offsetw2[7] = 0: Write DQ 270° delay amount <input type="checkbox"/> ctrl_offsetw2[5:0] x tFS	
PHY_CON8				
Address : 0xC00E_1020				
[31:24]	R/W	CTRL_OFFSETC3	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0: GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS	0x0
[23:16]	R/W	CTRL_OFFSETC2	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0: GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS	0x0
[15:8]	R/W	CTRL_OFFSETC1	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0: GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS	0x0
[7:0]	R/W	CTRL_OFFSETC0	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0: GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS	0x0
PHY_CON9				
Address : 0xC00E_1024				
[31:8]		RESERVED	Should be zero	0x0
[7:0]	R/W	CTRL_OFFSETC4 (40BIT)	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0: GATEout delay amount <input type="checkbox"/> ctrl_offsetc [6:0] x tFS	0x0
PHY_CON10				
Address : 0xC00E_1028				
[31:25]		RESERVED	Should be zero	0x0
[24]	R/W	CTRL_RESYNC	Active RISIG-EDGE signal. This signal should become LOW after set HIGH for normal operation. When this bit transits from LOW to HIGH, pointers of FIFO and DLL information is updated. In general, this bit should be set during initialization and refresh cycles. Refer to "MC-Initiated Update"(p105) to use ctrl_resync.	0x0
[23:8]		RESERVED	Should be zero	0x0
[7:0]	R/W	CTRL_OFFSETD	This field is for debug purpose. (For LPDDR2)	0x8

Bit	R/W	Symbol	Description	Reset Value
			If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. offset amount for 270° clock generation: ctrl_offsetd[7] = 1 : (tFS: fine step delay) 270° delay amount □ ctrl_offsetd[6:0] x tFS ctrl_offsetd[7] = 0 : 270° delay amount □ ctrl_offsetd[6:0] x tFS	
RESERVED				
Address : 0xC00E_102C				
PHY_CON12				
Address : 0xC00E_1030				
[31]		RESERVED	Should be zero	0x0
[30:24]	R/W	CTRL_START_POINT	Initial DLL lock start point. This is the number of delay cells and is the start point where "DLL" start tracing to be locked. Initial delay time is calculated by multiplying the unit delay of delay cell and this value.	7h10
[23]		RESERVED	Should be zero	0x0
[22:16]	R/W	CTRL_INC	Increase amount of start point	7h10
[14:8]	R/W	CTRL_FORCE	This field is used instead of ctrl_lock_value[9:2] found by the DLL only when ctrl_dll_on is LOW, i.e. If the DLL is off, this field is used to generate 270° clock and shift DQS by 90°.	0x0
[6]	R/W	CTRL_START	This field is used to start DLL locking.	0'b1
[5]	R/W	CTRL_DLL_ON	HIGH active start signal to turn on the DLL. This signal should be kept HIGH for normal operation. If this signal becomes LOW, DLL is turned off. This bit should be kept set before ctrl_start is set to turn on the DLL.	0'b1
[4:1]	R/W	CTRL_REF	This field determines the period of time when ctrl_locked is cleared. 4'b0000: Don't use. 4'b0001: ctrl_flock is de-asserted during 16 clock cycles, ctrl_locked is deasserted. 4'b0010: ctrl_flock is de-asserted during 24 clock cycles, ctrl_locked is deasserted. ~ 4'b1110: ctrl_flock is de-asserted during 120 clock cycles, ctrl_locked is deasserted. 4'b1111: Once ctrl_locked and dfi_init_complete are asserted, those won't be deasserted until rst_n is asserted.	4'h8
[0]		RESERVED	Should be zero	0x0
PHY_CON13				
Address : 0xC00E_1034				
[31:17]	R	RESERVED		-
[16:8]	R/W	CTRL_LOCK_VALUE	Locked delay line encoding value. ctrl_lock_value[8:2] : number of delay cells for coarse lock. ctrl_lock_value[1:0] : control value for fine lock. From ctrl_lock_value[8:0], tFS(fine step delay) can be calculated. tFS = tCK / ctrl_lock_value[9:0].	-
[2]	R	CTRL_CLOCK	Coarse lock information. According to clock jitter, ctrl_clock can be de-asserted.	-
[1]	R	CTRL_FLOCK	Fine lock information. According to clock jitter, ctrl_flock can be de-asserted.	-
[0]	R	CTRL_LOCKED	DLL stable lock information. This field is set after ctrl_flock is set. This field is cleared when ctrl_flock is de-asserted for some period of time specified by ctrl_ref[3:0] value. This field is required for stable lock status check.	-

Bit	R/W	Symbol	Description	Reset Value
PHY_CON14				
Address : 0xC00E_1038				
[31:12]	R	RESERVED		-
[11:8]	R/W	CTRL_PULLD_DQ	Active HIGH signal to down DQ signals. For normal operation this field should be zero. When bus is idle, it can be set to pull-down or up the bus not to make bus high-z state.	4'h0
[3:0]	R/W	CTRL_PULLD_DQS	Active HIGH signal to pull-up or down PDQS/NDQS signals. When using Gate Leveling in DDR3, this field can be zero. When bus is idle, it can be set to pull-down or up the bus not to make bus high-z state (PDQS/NDQS is pulled-down/up). For LPDDR2/LPDDR3 mode, this field should be always set for normal operation to make P/NDQS signals pull-down/up.	4'h0
PHY_CON15				
Address : 0xC00E_103C				
[31:21]	R	RESERVED		-
[20:16]	R	CTRL_FB_ERR	Feedback test stop with error for each slice. Ctrl_fb_err[0] : error of data channel0. Ctrl_fb_err[1] : error of data channel1. Ctrl_fb_err[2] : error of data channel2 for 32-bit PHY. Error of control for 16-bit PHY. Ctrl_fb_err[3] : error of data channel3 for 32-bit PHY. Ctrl_fb_err[4] : error of control channel for 32-bit PHY.	5'h0
[12:8]	R	CTRL_FB_OKAY	Feedback test completion without error for each slice. Ctrl_fb_oky[0] : okay of data channel0. Ctrl_fb_oky[1] : okay of data channel1. Ctrl_fb_oky[2] : okay of data channel2 for 32-bit PHY. Okay of control channel for 16-bit PHY. Ctrl_fb_oky[3] : okay of data channel3 for 32-bit PHY. Ctrl_fb_oky[4] : okay of control channel for 32-bit PHY.	5'h0
[4:0]	R/W	CTRL_FB_START	Feedback test start signal for each slice. Ctrl_fb_start[0]: start of data channel0 ctrl_fb_start[1]: start of data channel1 ctrl_fb_start[2]: start of data channel2 for 32-bit PHY start of control channel for 16-bit PHY ctrl_fb_start[3]: start of data channel3 for 32-bit PHY. Ctrl_fb_start[4]: start of control channel for 32-bit PHY. For this test, mode_highz should be set when memory is on the board	5'h0
PHY_CON16				
Address : 0xC00E_1040				
[31:28]	R/W	RESERVED	Should be zero	0x0
[27]	R/W	ZQ_CLK_EN	ZQ IO Clock enable	1'b1
[26:24]	R/W	ZQ_MODE_DDS	Driver strength selection.. It recommends one of the following settings instead of 3'h0. 3'b100 : 48Ω Impedance output driver 3'b101 : 40Ω Impedance output driver 3'b110 : 34Ω Impedance output driver 3'b111 : 30Ω Impedance output driver	3'h0
[23:21]	R/W	ZQ_MODE_TERM	On-die-termination(ODT) resistor value selection. "pb1pddr3_dls" and "pb1pddr3_dqs_dds" don't support ODT. 3'b001 : 120Ω Receiver termination 3'b010 : 60Ω Receiver termination 3'b011 : 40Ω Receiver termination 3'b100 : 30Ω Receiver termination	3'h0
[20]	R/W	ZQ_RGDDR3	GDDR3 mode enable signal(High: GDDR3 mode)	1'b0

Bit	R/W	Symbol	Description	Reset Value
[19]	R/W	ZQ_MODE_NOTERM	Termination disable selection. 1 : termination disable. 0 : termination enable. DDR : 1'b1 DDR2/DDR3 : 1b0(recommended) or 1'b1(when termination is not used) LPDDR2/LPDDR3 : 1b1 gDDR3 : 1b0	1'b0
[18]	R/W	ZQ_CLK_DIV_EN	Clock dividing enable	1'b0
[17:15]	R/W	ZQ_FORCE_IMPN	Immediate control code for pull-down.	3'h0
[14:12]	R/W	ZQ_FORCE_IMPP	Immediate control code for pull-up.	3'h0
[11:4]	R/W	ZQ_UDT_DLY	ZQ I/O clock enable duration for auto calibration mode.	8'h30
[3:2]	R/W	ZQ_MANUAL_MODE	Manual calibration mode selection 2'b00: force calibration 2'b01: long calibration 2;b10: short calibration	2'b01
[1]	R/W	ZQ_MANUAL_STR	Manual calibration start	1'b0
[0]	R/W	ZQ_AUTO_EN	Auto calibration enable	1'b0
PHY_CON17				
Address : 0xC00E_1048				
[31:9]	R/W	RESERVED	Should be zero	-
[8:6]	R	ZQ_PMON	Control code found by auto calibration for pull-up.	-
[5:3]	R	ZQ_NMON	Control code found by auto calibration for pull-down.	-
[2]	R	ZQ_ERROR	Calibration fail indication (High: calibration failed)	-
[1]	R	ZQ_PENDING	Auto calibration enable status	-
[0]	R	ZQ_DONE	ZQ Calibration is finished.	1'b0
PHY_CON18				
Address : 0xC00E_104C				
[31:28]	R	RESERVED		0x0
[27:24]	R	DM_FAIL_STATUS	It will be enabled if there is no pass period after DM Calibration. It should be read by zero if Calibration is done normally. Please refer to ReadModeCon(=PHY_CON5) to read.	0x0
[23:18]	R	T3_RDDATA_EN	Trddata_en timing parameter for data slice 3. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15
[17:12]	R	T2_RDDATA_EN	Trddata_en timing parameter for data slice 2. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15
[11:6]	R	T1_RDDATA_EN	Trddata_en timing parameter for data slice 1. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15
[5:0]	R	T0_RDDATA_EN	Trddata_en timing parameter for data slice 0. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15
PHY_CON19				
Address : 0xC00E_1050				
[31:24]	R	RDLVL_OFFSETR3	DQ Calibration SDLL code for data slice 3. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[23:16]	R	RDLVL_OFFSETR2	DQ Calibration SDLL code for data slice 2. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[15:8]	R	RDLVL_OFFSETR1	DQ Calibration SDLL code for data slice 1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[7:0]	R	RDLVL_OFFSETR0	DQ Calibration SDLL code for data slice 0. Please refer to ReadModeCon(=PHY_CON5) to read.	-
PHY_CON20				
Address : 0xC00E_1054				
[31:24]	R	RDLVL_OFFSETC3	Gate Training SDLL code for data slice 3. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[23:16]	R	RDLVL_OFFSETC2	Gate Training SDLL code for data slice 2. Please refer to ReadModeCon(=PHY_CON5) to read.	-

Bit	R/W	Symbol	Description	Reset Value
[15:8]	R	RDLVL_OFFSETC1	Gate Training SDLL code for data slice 1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[7:0]	R	RDLVL_OFFSETC0	Gate Training SDLL code for data slice 0. Please refer to ReadModeCon(=PHY_CON5) to read.	-
PHY_CON21				
Address : 0xC00E_1058				
[31:0]	R	VWM_FAIL_STATUS	It will be enabled if there is no pass period after DQ Calibration. It should be read by zero if Calibration is done normally. Please refer to ReadModeCon(=PHY_CON5) to read.	0x0
PHY_CON22				
Address : 0xC00E_105C				
[31:20]	R	RESERVED		-
[19:0]	R/W	LPDDR2_ADDR	<p>LPDDR2/LPDDR3 Address. Default value(=0x208) is Mode Register Reads to DQ Calibration registers MR32. Reads to MR32 return DQ Calibration Pattern "1111-0000-1111-0000" on DQ[0], DQ[8], DQ[16], and DQ[24] for x32 devices. When doing Write Training, This field should be set by READ command. For example, lpddr2_addr2 will be 20h5 if the column address is 11h0 and bank address is 3b000.</p> <p>According to READ Command definition in LPDDR2 or LPDDR3 lpddr2_addr[19:0] = "C11-C10-C9-C8-C7-C6-C5-C4-C3-AP-BA2-BA1-BA0-C2-C1-R-R-H-L-H" (C means Column Address, BA means Bank Address, R means Reserved)</p> <p>In case of CA swap mode, lpddr2_addr=20h41 for Read Training and lpddr2_addr=20h204 for Write Training if the column address is 11h0 and bank address is 3b000.</p> <p>lpddr2_addr[19:0] = "AP-C3-C9-C5-C6-C7-C8-C4-C10-C11-H-L-BA0-R-R-C1-C2-H-B1-B2"(CA swap mode)</p>	0x208
PHY_CON23				
Address : 0xC00E_1060				
[31:20]	R	RESERVED		-
[19:0]	R/W	LPDDR2_DEFAULT	LPDDR2/LPDDR3 Default Address	0x3ff
PHY_CON24				
Address : 0xC00E_1064				
[31:16]	R/W	DDR3_DEFAULT	DDR3 Default Address	0x0
[15:0]	R/W	DDR3_ADDR	DDR3 Address	0x0
[0]	R/W	CA_SWAP_MODE	<p>If ctrl_ddr_mode[1]=1 and ca_swap_mode=1, PHY will be in "CA swap mode" for POP. In "CA swap mode", CA[9:0] will be swapped in the following way. CA[0] □ CA[9] CA[1] □ CA[8] CA[7] CA[3] □ CA[6] CA[4] □ CA[5] CA[5] □ CA[4] CA[6] □ CA[3] CA[7] □ CA[1] CA[8] □ CA[1] CA[9] □ CA[0]</p> <p>NOTE: Don't use "ctrl_atgate=1" in normal operation when ca_swap_mode = 1. "ctrl_atgate" can be enabled only during calibration.</p>	0x0
PHY_CON25				
Address : 0xC00E_1068				
[31:30]	R	RESERVED		-
[29:16]	R/W	DDR3_CMD	DDR3 Command	0x105E
[15:14]	R	RESERVED		-
[13:0]	R/W	LPDDR2_CMD	LPDDR2/3 Command. This field should be "16'h000E" using LPDDR2 or LPDDR3.	0x107E
PHY_CON26				
Address : 0xC00E_106C				
[31:21]	R	RESERVED		-
[20:16]	R/W	T_WRDATA_EN	Clock cycles between write command and the first edge of DQS which can capture the first valid DQ. For example, if WL is 6, it should be set as 7(=WL+1) in LPDDR3, 6(=WL) in DDR3.	0x8

Bit	R/W	Symbol	Description	Reset Value
[13:0]	R/W	CMD_DEFAULT	Default Command 16'h000F(LPDDR2, LPDDR3) 16'h107F(DDR2, DDR3)	0x107F
PHY_CON27				
Address : 0xC00E_1070				
[31:24]	R	RLVL_VWML3	Left Code Value in Read Valid Window Margin for Data Slice3. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[23:16]	R	RLVL_VWML2	Left Code Value in Read Valid Window Margin for Data Slice2. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[15:8]	R	RLVL_VWML1	Left Code Value in Read Valid Window Margin for Data Slice1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[7:0]	R	RLVL_VWML0	Left Code Value in Read Valid Window Margin for Data Slice0. Please refer to ReadModeCon(=PHY_CON5) to read.	-
PHY_CON28				
Address : 0xC00E_1074				
[31:24]	R	RLVL_VWMR3	Right Code Value in Read Valid Window Margin for Data Slice3. Please refer to ReadModeCon(=PHY_CON5) to read	-
[23:16]	R	RLVL_VWMR2	Right Code Value in Read Valid Window Margin for Data Slice2. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[15:8]	R	RLVL_VWMR1	Right Code Value in Read Valid Window Margin for Data Slice1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
[7:0]	R	RLVL_VWMR0	Right Code Value in Read Valid Window Margin for Data Slice0. Please refer to ReadModeCon(=PHY_CON5) to read.	-
PHY_CON29				
Address : 0xC00E_1078				
[31:0]	R	VERSION_INFO	Version Information	0x0501_0203
PHY_CON30				
Address : 0xC00E_107C				
[30:24]	R/W	CTRL_WRLVL3_CODE	Write Level Slave DLL Code Value for Data_Slice 3(0x8~0x38)	0x0
[23:17]	R/W	CTRL_WRLVL2_CODE	Write Level Slave DLL Code Value for Data_Slice 2(0x8~0x38)	0x0
[16]	R/W	CTRL_WRLVL	Write Level Enable	0x0
[15]	R	RESERVED		-
[14:8]	R/W	CTRL_WRLVL1_CODE	Write Level Slave DLL Code Value for Data_Slice 1(0x8~0x38)	0x0
[7]	R	RESERVED		-
[6:0]	R/W	CTRL_WRLVL0_CODE	Write Level Slave DLL Code Value for Data_Slice 0(0x8~0x38)	0x0
PHY_CON31				
Address : 0xC00E_1080				
[31:28]	R/W	CA4DESKEWCODE	DeSkew Code for CA[4] (0x8~0x60)	0x0
[27:21]	R/W	CA3DESKEWCODE	DeSkew Code for CA[3] (0x8~0x60)	0x0
[20:14]	R/W	CA2DESKEWCODE	DeSkew Code for CA[2] (0x8~0x60)	0x0
[13:7]	R/W	CA1DESKEWCODE	DeSkew Code for CA[1] (0x8~0x60)	0x0
[6:0]	R/W	CA0DESKEWCODE	DeSkew Code for CA[0] (0x8~0x60)	0x0
PHY_CON32				
Address : 0xC00E_1084				
[31]	R/W	CA9DESKEWCODE	DeSkew Code for CA[9] (0x8~0x60)	0x0

Bit	R/W	Symbol	Description	Reset Value
[30:24]	R/W	CA8DESKEWCODE	DeSkew Code for CA[8] (0x8~0x60)	0x0
[23:17]	R/W	CA7DESKEWCODE	DeSkew Code for CA[7] (0x8~0x60)	0x0
[16:10]	R/W	CA6DESKEWCODE	DeSkew Code for CA[6] (0x8~0x60)	0x0
[9:3]	R/W	CA5DESKEWCODE	DeSkew Code for CA[5] (0x8~0x60)	0x0
[2:0]	R/W	CA4DESKEWCODE	DeSkew Code for CA[4] (0x8~0x60)	0x0
PHY_CON33				
Address : 0xC00E_1088				
[31:27]	R/W	CKE0DESKEWCODE	DeSkew Code for CKE[0], (0x8~0x60)	0x0
[26:20]	R/W	CS1DESKEWCODE	DeSkew Code for CS[1] (0x8~0x60)	0x0
[19:13]	R/W	CS0DESKEWCODE	DeSkew Code for CS[0] (0x8~0x60)	0x0
[12:6]	R/W	CKDESKEWCODE	DeSkew Code for CK, (0x8~0x60)	0x0
[5:0]	R/W	CA9DESKEWCODE	DeSkew Code for CA[9] (0x8~0x60)	0x0
PHY_CON34				
Address : 0xC00E_108C				
[31:9]	R	RESERVED		-
[8:2]	R/W	CKE1DESKEWCODE	DeSkew Code for CKE[1] (0x8~0x60)	0x0
[1:0]	R/W	CKE0DESKEWCODE	DeSkew Code for CKE[0] (0x8~0x60)	0x0
RESERVED				
Address : 0xC00E_1090 ~ 0xC00E_1094				
PHY_CON37				
Address : 0xC00E_1098				
[31:6]	R	RESERVED		-
[5:0]	R/W	RSTDESKEWCODE	DeSkew Code for Reset (0x8~0x38)	0x0
RESERVED				
Address : 0xC00E_109C				
PHY_CON39				
Address : 0xC00E_10A0				
[31:28]	R	RESERVED		-
[27:25]	R/W	DA3DS	Driver Strength Selection for Data Slice 3	0x0
[24:22]	R/W	DA2DS	Driver Strength Selection for Data Slice 2	0x0
[21:19]	R/W	DA1DS	Driver Strength Selection for Data Slice 1	0x0
[18:16]	R/W	DA0DS	Driver Strength Selection for Data Slice 0	0x0
[11:9]	R/W	CACKDRVRS	Driver Strength Selection for CK	0x0
[8:6]	R/W	CACKEDRVRS	Driver Strength Selection for Cke[1:0]	0x0
[5:3]	R/W	CACSDRVRS	Driver Strength Selection for CS[1:0]	0x0
[2:0]	R/W	CAADRDRVRS	Driver Strength Selection for ADCT[15:0].	0x0
PHY_CON40				
Address : 0xC00E_10A4				
[31:0]	R/W	CTRL_ZQ_CLK_DIV	ZQ Clock divider setting value	0x7

Bit	R/W	Symbol	Description	Reset Value
PHY_CON41				
Address : 0xC00E_10A8				
[31:0]	R/W	CTRL_ZQ_TIMER	It controls the interval between each ZQ calibration	0xF0
PHY_CON42				
Address : 0xC00E_10AC				
[31:13]	R	RESERVED		-
[12:8]	R/W	CTRL_BSTLEN	Burst Length(BL)	5h0
[7:5]	R/W	RESERVED		-
[4:0]	R/W	CTRL_RDLAT	Read Latency(RL)	5h0

14.3.3 MCU-S Memory Control Register

Bit	R/W	Symbol	Description	Reset Value
MEMORY BUS WIDTH REGISTER (MEMBW)				
Address : 0XC005_1000				
[31]	R/W	RESERVED	Reserved	1'b0
[30]	R/W	INTSRAMSHADOW1	Base Address of Internal ROM 0 : Bass address = 0x0000_0000 1 : Bass address = 0x3400_0000	CfgBootMode
[29:2]	R/W	RESERVED	Reserved	1'b0
[1]	R/W	SR1BW	Set data bus width of static #1 0: Byte (8bit) 1: Half-word (16bit)	1'b0
[0]	R/W	SR0BW	Set data bus width of static #0 0: Byte (8bit) 1: Half-word (16bit)	CfgSTBUSWidth
MEMORY TIMING FOR TACS LOW REGISTER (MEMTIMEACSL)				
Address : 0XC005_1004				
[31:8]	R/W	RESERVED	Reserved	24'bx
[7:4]	R/W	TACS1	tACS of static #1 tACS = TACS + 1	4'b11
[3:0]	R/W	TACSO	tACS of static #0 tACS = TACS + 1 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	CfgBootMode
MEMORY TIMING FOR TACS HIGH REGISTER (MEMTIMEACSH)				
Address : 0XC005_1008				
[31:24]	R/W	RESERVED	Reserved	8'b0
[23:19]	R/W	TACS13	tACS of static #13 tACS = TACS + 1 (Unit : BCLK)	4'b0
[19:0]	R/W	RESERVED	reserved	x

Bit	R/W	Symbol	Description	Reset Value
MEMORY TIMING FOR TCOS LOW REGISTER (MEMTIMECOSL)				
Address : 0XC005_100C				
[31:8]	R/W	RESERVED	reserved	x
[7:4]	R/W	TCOS1	tCOS of static #1 tCOS = TCOS + 1 (Unit : BCLK)	4'b11
[3:0]	R/W	TCOS0	tCOS of static #0 tCOS = TCOS + 1 (Unit : BCLK) 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	CfgBootMode
MEMORY TIMING FOR TCOS HIGH REGISTER (MEMTIMECOSH)				
Address : 0XC005_1010				
[31:24]	R/W	RESERVED	Reserved	x
[23:20]	R/W	TCOS13	tCOS of static #13 tCOS = TCOS + 1 (Unit : BCLK)	4'b0
[19]	R/W	TCOS8	tCOS of static #8 tCOS = TCOS + 1 (Unit : BCLK)	x
MEMORY TIMING FOR TACC 0 REGISTER (MEMTIMEACC0)				
Address : 0XC005_1014				
[31:16]	R/W	RESERVED	Reserved	x
[15:8]	R/W	TACC1	tACC of static #1	8'b01000
[7:0]	R/W	TACC0	tACC of static #0	CfgBootMode
MEMORY TIMING FOR TACC 1 REGISTER (MEMTIMEACC1)				
Address : 0XC005_1018				
[31:0]	R/W	RESERVED	Reserved	x
MEMORY TIMING FOR TACC 2 REGISTER (MEMTIMEACC2)				
Address : 0XC005_101C				
[31:0]	R/W	RESERVED	Reserved	x
MEMORY TIMING FOR TACC 3 REGISTER (MEMTIMEACC3)				
Address : 0XC005_1020				
[31:16]	R/W	RESERVED	Reserved	x
[15:8]	R/W	TACC13	tACC of static #13	8'b0
[7:0]	R/W	RESERVED	Reserved	x
MEMORY TIMING FOR TSACC 0 REGISTER (MEMTIMESACC0)				
Address : 0XC005_1024				
[31:16]	R/W	RESERVED	Reserved	x
[15:8]	R/W	TSACC1	tSACC of static #1	8'b0
[7:0]	R/W	TSACC0	tSACC of static #0	8'b0

Bit	R/W	Symbol	Description	Reset Value
MEMORY TIMING FOR TSACC 1 REGISTER (MEMTIMESACC1)				
Address : 0XC005_1028				
[31:0]	R/W	RESERVED	Reserved	x
MEMORY TIMING FOR TSACC 2 REGISTER (MEMTIMESACC2)				
Address : 0XC005_102C				
[31:0]	R/W	RESERVED	Reserved	x
MEMORY TIMING FOR TSACC 3 REGISTER (MEMTIMESACC3)				
Address : 0XC005_1030				
[31:15]	R/W	RESERVED	Reserved	x
[15:8]	R/W	TSACC13	tSACC of static #13	8b0
[7:0]	R/W	RESERVED	Reserved	x
RESERVED				
Address : 0XC005_1034 ~0xC005_1040				
MEMORY TIMING FOR TCOH LOW REGISTER (MEMTIMECOHL)				
Address : 0XC005_1044				
[31:8]	R/W	RESERVED	Reserved	x
[7:4]	R/W	TCOH1	tCOH of static #1	4b11
[3:0]	R/W	TCOH0	tCOH of static #0 tCOH = TCOH + 1 (Unit : BCLK) 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	CfgBootMode
MEMORY TIMING FOR TCOH HIGH REGISTER (MEMTIMECOHH)				
Address : 0XC005_1048				
[31:24]	R/W	RESERVED	Reserved	8b0
[23:20]	R/W	TCOH13	tCOH of static #13.	4b0
[19:0]	R/W	RESERVED	Reserved	x
MEMORY TIMING FOR TCAH LOW REGISTER (MEMTIMECAHL)				
Address : 0XC005_104C				
[31:8]	R/W	RESERVED	Reserved	x
[7:4]	R/W	TCAH1	tCAH of static #1	4b11
[3:0]	R/W	TCAH0	tCAH of static #0 tCAH = TCAH + 1 (Unit : BCLK) 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	CfgBootMode
MEMORY TIMING FOR TCAH HIGH REGISTER (MEMTIMECAHH)				
Address : 0XC005_1050				

Bit	R/W	Symbol	Description	Reset Value
[31:24]	R/W	RESERVED	Reserved	x
[23:20]	R/W	TCAH13	tCAH of static #13	4'b0
[19:0]	R/W	RESERVED	Reserved	x
MEMORY BURST CONTROL LOW REGISTER (MEMBURSTL)				
Address : 0XC005_1054				
[9:8]	R/W	RESERVED	Reserved	x
[7:6]	R/W	BWRITE1	Write Access Control of static #1 00: Disable 01: 4 byte burst Access 10: 8 byte burst Access 11: 16 byte burst Access	2b00
[5:4]	R/W	BREAD1	Read Access Control of static #1 00: Disable 01: 4 byte burst Access 10: 8 byte burst Access 11: 16 byte burst Access	2b01
[3:2]	R/W	BWRITE0	Write Access Control of static #0 00: Disable 01: 4 byte burst Access 10: 8 byte burst Access 11: 16 byte burst Access	2b00
[1:0]	R/W	BREAD0	Read Access Control of static #0 00: Disable 01: 4 byte burst Access 10: 8 byte burst Access 11: 16 byte burst Access	2b00
RESERVED				
Address : 0XC005_1058				
[31:0]	R/W	RESERVED	Must be low	0
MEMORY WAIT CONTROL REGISTER (MEMWAIT)				
Address : 0XC005_105C				
[31:4]	R/W	RESERVED	Must be low	'b0
[3]	R/W	WAITENB1	Wait Enable control of static #1 0: Disable Wait Control 1: Enable Wait Control	1'b0
[2]	R/W	WAITPOL1	Wait Polarity control of static #1 If it is set to low active wait signal, the wait is enabled when the input signal is low. If it is set to high active wait signal, the wait is enabled when the input signal is high. 0: High active wait signal 1: Low active wait signal	1'b0
[1]	R/W	WAITENB0	Wait Enable control of static #0 0: Disable Wait Control 1: Enable Wait Control	1'b0
[0]	R/W	WAITPOLO	Wait Polarity control of static #0 If it is set to low active wait signal, the wait is enabled when the input signal is low. If it is set to high active wait signal, the wait is enabled when the input signal is high. 0: High active wait signal 1: Low active wait signal	1'b0
RESERVED				
Address : 0XC005_1060 ~ 0xC005_1084				
NAND FLASH CONTROL REGISTER (NFCONTROL)				
Address : 0XC005_1088				
[31]	R/W	NCSEN _B	nNCS Enable 0: Disbale 1 : Enable	1'b0

Bit	R/W	Symbol	Description	Reset Value
[30]	R/W	NFECCAUTORSTENB	0 : Auto Reset Disable 1 : Auto Reset Enable	1'b1
[29:27]	R/W	NFECCMODE	Number of Error correction bit 0 : 4bit 1 : 8bit 2 : 12bit 3 : 16 bit 4 : 24 bit(Only data Size 1024byte) 5 : 24bis 6 : 40 bits(Only data Size 1024byte) 7 : 60bits(Only data Size 1024byte)	3'b000
[26:16]	R	RESERVED	Reserved	11'b0
[15]	R/W	IRQPEND	Interrupt pending bit of RnB signal detect. Read : 0 : Not pended Write : 0 : No affect	1'bx
[14:12]	R	RESERVED	Reserved	3'h0
[11]	W	ECCRST	HW ECC block reset NFECL, NFECH, NFCNT, NFECCSTATUS, NFSYNDRONE31/75 Registers Reset	1'b0
[10]	R	RESERVED	Reserved	1'bx
[9]	R	RNB	Ready/Busy check of NAND Flash operation. 0: Busy 1 : Ready	1'bx
[8]	R/W	IRQENB	Set interrupt enable/disable at the rising edge of RnB signal of NAND Flash. 0 : Disable 1 : Enable	1'b0
[7]	R	RESERVED	Reserved	1'b0
[6:5]	R/W	NFCARTRIDEENB	Set NAND Flash Cartridge Enable Used SDEX bus of NAND access. write 00 : Disable 01 : Enable read 00 : disable 10 : enable	1'b0
[4:3]	R/W	NFTYPE	Set NAND Flash Type for NAND Booting. 00 : Small block 3 address NAND 01 : Small block 4 address NAND 10 : Large block 4 address NAND 11 : Large block 5 address NAND	CfgNFType
[2]	R/W	RESERVED	Reserved	1'b0
[1:0]	R/W	NFBANK	Set NAND Flash bank for access. This bit determines which one will be selected out of nNCS[2:0]. Selected nNCS applied after NFBANK is changed and static memory is accessed. 0 : nNCS[0] 1 : nNCS[1] 2 : nNCS[2] 3 : reserved	2'b0

NAND ECC CONTROL REGISTER (NFECCCTRL)

Address : 0XC005_108C

[31:29]	R	RESERVED	Reserved	-
[28]	R	ERROR	Result of Decode 0 : No Error 1 : Error	1'b0
[28]	W	NUMBEROFERROR	Number Of Error	5'h0
[24]	R/W	DECMode	0 : Encoder 1 : Decoder	1'b0
[27]	W	LOADELP	Load ELP Register	-
[26]	W	DECMode	0 : Encoder 1 : Decoder	1'b0
[25]	R/W	RESERVED	Must be zero	-
[24:18]	R/W	NUMBEROFLP	Number Of ELP	5'h0

Bit	R/W	Symbol	Description	Reset Value
			When NFECCMODE Register = 0 then 4 When NFECCMODE Register = 1 then 8 When NFECCMODE Register = 2 then 12 When NFECCMODE Register = 3 then 16 When NFECCMODE Register = 4 then 24	
[17:10]	R/W	PARITYCONUT	Number of Parity byte When NFECCMODE Register = 0 then 5 When NFECCMODE Register = 1 then 12 When NFECCMODE Register = 2 then 18 When NFECCMODE Register = 3 then 25 When NFECCMODE Register = 4 then 41	6h0
[9:0]	R/W	NFCOUNTVALUE	Number of NAND Read and Write data When NFECCMODE Register = 0~3 then 512 When NFECCMODE Register = 4 then 1024	10'h0
NAND FLASH DATA COUNT REGISTER (NFCNT)				
Address : 0XC005_1090				
[31:26]	R	RESERVED	Reserved	6h0
[25:16]	R	NFWRCNT	NAND Flash Write Data Count	-
[15:10]	R	RESERVED	Reserved	6h0
[9:0]	R	NFRDCNT	NAND Flash Read Data Count	-
NAND FLASH ECC STATUS REGISTER (NFECCSTATUS)				
Address : 0XC005_1094				
[31:3]	R	RESERVED	Reserved	29'h0
[2]	R	NFCHECKERROR	When completing NAND Read operating, error check on Read Data. If Read Data Error occurs, NFCHECKERROR is set as '1'. Then NAND Address/Command writes, the register is cleared. 0 : No Error 1 : Data Error	1b0
[1]	R	NFECCDEC DONE	When reading NAND Data with 512 byte plus 51 Cycles (BCLK), NFECCDEC DONE is set as '1'. When NAND Address/Command writes, it's cleared. 0 : IDLE or RUN 1 : End	1b0
[0]	R	NFECCENCDONE	After writing NAND DATA with 512 byte, NFECCENCDONE is set as '1'. When reading ECC Register Data, it's cleared. 0 : IDLE or RUN 1 : End	1b0
NAND TIMING FOR TACS REGISTER (NFTIMEACS)				
Address : 0XC005_1098				
[31:12]	R/W	RESERVED	Reserved	x
[11:8]	R/W	TNFACS2	NAND tNFACS of Nand Bank 2	4'h0
[7:4]	R/W	TNFACS1	NAND tNFACS of Nand Bank 1	4'h0
[3:0]	R/W	TNFACSO	NAND tNFACS of Nand Bank 0 tNFACS = TNFACS + 1 (Unit : BCLK) 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	4'h7

Bit	R/W	Symbol	Description	Reset Value
NAND TIMING FOR TCOS REGISTER (NFTIMECOS)				
Address : 0XC005_109C				
[31:12]	R/W	RESERVED	Reserved	x
[11:8]	R/W	TNF COS2	NAND tNFCOS of Nand Bank 2	4'h0
[7:4]	R/W	TNF COS1	NAND tNFCOS of Nand Bank 1	4'h0
[3:0]	R/W	TNF COS0	NAND tNFCOS of Nand Bank 0 tNFCOS = TNFCOS + 1 (Unit : BCLK) 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	4'h7
NAND TIMING FOR TACC0 REGISTER (NFTIMEACCO)				
Address : 0XC005_10A0				
[31:24]	R/W	RESERVED	Reserved	x
[23:16]	R/W	TNF ACC2	tANFCC of NAND Bank 2	8'h00
[15:8]	R/W	TNF ACC1	tANFCC of NAND Bank 1	8'h00
[7:0]	R/W	TNF ACC0	tANFCC of NAND Bank 0 tNFACC = TNFACC + 1 Cycle	8'h0f
RESERVED				
Address : 0XC005_10A4				
NAND TIMING FOR TOCH REGISTER (NFTIMEOCH)				
Address : 0XC005_10A8				
[31:12]	R/W	RESERVED	Reserved	x
[11:8]	R/W	TNF OCH2	NAND tINFOCH of Nand Bank 2	4'h0
[7:4]	R/W	TNF OCH1	NAND tINFOCH of Nand Bank 1	4'h0
[3:0]	R/W	TNF OCH0	NAND tINFOCH of Nand Bank 0 tINFOCH = TNFOCH + 1 (Unit : BCLK) 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	4'h7
NAND TIMING FOR TCAH REGISTER (NFTIMECAH)				
Address : 0XC005_10AC				
[31:12]	R/W	RESERVED	Reserved	x
[11:8]	R/W	TNF CAH2	NAND tNFCAH of Nand Bank 2	4'h0
[7:4]	R/W	TNF CAH1	NAND tNFCAH of Nand Bank 1	4'h0
[3:0]	R/W	TNF CAH0	NAND tNFCAH of Nand Bank 0 tNFCAH = TNFCAH + 1 (Unit : BCLK) 0000: 1 cycle 0001: 2 cycle 0010: 3 cycle 0011: 4 cycle 0100: 5 cycle 0101: 6 cycle 0110: 7 cycle 0111: 8 cycle 1000: 9 cycle 1001: 10 cycle 1010: 11 cycle 1011: 12 cycle 1100: 13 cycle 1101: 14 cycle 1110: 15 cycle 1111: 0 cycle	4'h7

Bit	R/W	Symbol	Description	Reset Value
NAND FLASH ECC 0 ~ 26 REGISTER (NFECC0 ~ NFECC26)				
Address : 0XC005_10B0 ~ 0XC005_1118				
[31:0]	R	ECC[N]	Represents 512/1024 byte ECC parity code during Write operation by H/W ECC generation block	32'h0
<Note> n=0 ~ 26 ECC0 : 0XC005_1000 + 0x0B0 ECC1 : 0XC005_1000 + 0x0B4 ECC2 : 0XC005_1000 + 0x0B8 ECC3 : 0XC005_1000 + 0x0BC ECC4 : 0XC005_1000 + 0x0C0 ECC5 : 0XC005_1000 + 0x0C4 ECC6 : 0XC005_1000 + 0x0C8 ECC7 : 0XC005_1000 + 0x0CC ECC8 : 0XC005_1000 + 0x0D0 ECC9 : 0XC005_1000 + 0x0D4 ECC10 : 0XC005_1000 + 0x0D8 ECC11 : 0XC005_1000 + 0x0DC ECC12 : 0XC005_1000 + 0x0E0 ECC13 : 0XC005_1000 + 0x0E4 ECC14 : 0XC005_1000 + 0x0E8 ECC15 : 0XC005_1000 + 0x0EC ECC16 : 0XC005_1000 + 0x0F0 ECC17 : 0XC005_1000 + 0x0F4 ECC18 : 0XC005_1000 + 0x0F8 ECC19 : 0XC005_1000 + 0x0FC ECC20 : 0XC005_1000 + 0x100 ECC21 : 0XC005_1000 + 0x104 ECC22 : 0XC005_1000 + 0x108 ECC23 : 0XC005_1000 + 0x10C ECC24 : 0XC005_1000 + 0x110 ECC25 : 0XC005_1000 + 0x114 ECC26 : 0XC005_1000 + 0x118				
NAND FLASH ORIGIN ECC 0~26 REGISTER (NFORGECC0 ~ NFORGECC26)				
Address : 0XC005_111C ~ 0XC005_1184				
[31:0]	R/W	ORGECC[N]	When NAND Read operates, firstly read Original ECC Data already saved in the NAND Spare area and then writes in Register.	32'h0
<Note> n=0 ~ 26 ORGECC0 : 0XC005_1000 + 0x11C ORGECC1 : 0XC005_1000 + 0x120 ORGECC2 : 0XC005_1000 + 0x124 ORGECC3 : 0XC005_1000 + 0x128 ORGECC4 : 0XC005_1000 + 0x12C ORGECC5 : 0XC005_1000 + 0x130 ORGECC6 : 0XC005_1000 + 0x134 ORGECC7 : 0XC005_1000 + 0x138 ORGECC8 : 0XC005_1000 + 0x13C ORGECC9 : 0XC005_1000 + 0x140 ORGECC10 : 0XC005_1000 + 0x144 ORGECC11 : 0XC005_1000 + 0x148 ORGECC12 : 0XC005_1000 + 0x14C ORGECC13 : 0XC005_1000 + 0x150 ORGECC14 : 0XC005_1000 + 0x154 ORGECC15 : 0XC005_1000 + 0x158 ORGECC16 : 0XC005_1000 + 0x15C ORGECC17 : 0XC005_1000 + 0x160 ORGECC18 : 0XC005_1000 + 0x164 ORGECC19 : 0XC005_1000 + 0x168 ORGECC20 : 0XC005_1000 + 0x16C ORGECC21 : 0XC005_1000 + 0x170 ORGECC22 : 0XC005_1000 + 0x174 ORGECC23 : 0XC005_1000 + 0x178 ORGECC24 : 0XC005_1000 + 0x17C				

Bit	R/W	Symbol	Description	Reset Value
ORGECC25 : 0XC005_1000 + 0x180				
ORGECC26 : 0XC005_1000 + 0x184				
NAND FLASH ECC SYNDROME VALUE 0~29 REGISTER (NFSYNDROME0 ~ NFSYNDROME29)				
Address : 0XC005_1188 ~ 0XC005_11FC				
[31:30]	R	RESERVED	Reserved	2'b0
[29:16]	R	SYNDROM[3+(NX4)]	ECC Decoder Result Odd Syndrome Data3(offset n x 4)	14'h0
[15:14]	R	RESERVED	Reserved	2b0
[13:0]	R	SYNDROM[1+(NX4)]	ECC Decoder Result Odd Syndrome Data1 (offset n x 4)	14'h0
<Note> n = 0~29				
NFSYNDROME0 : SYNDROME3/1 0XC005_1000 + 0x188				
NFSYNDROME1 : SYNDROME7/5 0XC005_1000 + 0x18C				
NFSYNDROME2 : SYNDROME11/9 0XC005_1000 + 0x190				
NFSYNDROME3 : SYNDROME15/13 0XC005_1000 + 0x194				
NFSYNDROME4 : SYNDROME19/17 0XC005_1000 + 0x198				
NFSYNDROME5 : SYNDROME23/21 0XC005_1000 + 0x19C				
NFSYNDROME6 : SYNDROME27/25 0XC005_1000 + 0x1A0				
NFSYNDROME7 : SYNDROME31/29 0XC005_1000 + 0x1A4				
NFSYNDROME8 : SYNDROME35/33 0XC005_1000 + 0x1A8				
NFSYNDROME9 : SYNDROME39/37 0XC005_1000 + 0x1AC				
NFSYNDROME10 : SYNDROME43/41 0XC005_1000 + 0x1B0				
NFSYNDROME11 : SYNDROME47/45 0XC005_1000 + 0x1B4				
NFSYNDROME12 : SYNDROME51/49 0XC005_1000 + 0x1B8				
NFSYNDROME13 : SYNDROME55/53 0XC005_1000 + 0x1BC				
NFSYNDROME14 : SYNDROME59/57 0XC005_1000 + 0x1C0				
NFSYNDROME15 : SYNDROME63/61 0XC005_1000 + 0x1C4				
NFSYNDROME16 : SYNDROME67/65 0XC005_1000 + 0x1C8				
NFSYNDROME17 : SYNDROME71/69 0XC005_1000 + 0x1CC				
NFSYNDROME18 : SYNDROME75/73 0XC005_1000 + 0x1D0				
NFSYNDROME19 : SYNDROME79/77 0XC005_1000 + 0x1D4				
NFSYNDROME20 : SYNDROME83/81 0XC005_1000 + 0x1D8				
NFSYNDROME21 : SYNDROME87/85 0XC005_1000 + 0x1DC				
NFSYNDROME22 : SYNDROME91/89 0XC005_1000 + 0x1E0				
NFSYNDROME23 : SYNDROME95/93 0XC005_1000 + 0x1E4				
NFSYNDROME24 : SYNDROME99/97 0XC005_1000 + 0x1E8				
NFSYNDROME25 : SYNDROME103/101 0XC005_1000 + 0x1EC				
NFSYNDROME26 : SYNDROME107/105 0XC005_1000 + 0x1F0				
NFSYNDROME27 : SYNDROME111/109 0XC005_1000 + 0x1F4				
NFSYNDROME28 : SYNDROME115/113 0XC005_1000 + 0x1F8				
NFSYNDROME29 : SYNDROME119/117 0XC005_1000 + 0x1FC				
NAND FLASH ECC ELP VALUE 0 ~ 29 REGISTER (NFELP0 ~ NFELP29)				
Address : 0XC005_1200 ~ 0XC005_1274				
[31:28]	R/W	RESERVED	Reserved	-
[27:14]	R/W	ELP[2+(NX2)]	2 nd ELP(Error locator polynomial) register	-
[13:0]	R/W	ELP[1+(NX2)]	1 st t ELP(Error locator polynomial) register	-
NAND FLASH ERROR LOCATION 0 ~ 59 REGISTER (NFERRORLOCATION0 ~ NFERRORLOCATION59)				
Address : 0XC005_1278 ~ 0XC005_12EC				
[31:28]	R	RESERVED	Reserved	-
[27:14]	R	ERROR[2+(NX2)]	2 nd location of error	-
[13:0]	R	ERROR[1+(NX2)]	1 st location of error	-

Bit	R/W	Symbol	Description	Reset Value
NAND FLASH ERROR LOCATION 0 ~ 59 REGISTER (NFERRORLOCATION0 ~ NFERRORLOCATION59)				
Address : 0XC005_12F0				
[31:2]	R	RESERVED	Reserved	
[1]	W	CPUSYND	User can write the SYNDROM value or auto SYNDROM value. 0 : auto 1 : user write	1'b0
[0]	W	CPUELPM	User can write the ELP value or auto ELP value. 0 : auto 1 : user write	1'b0
NAND FLASH ECC WRITE SYNDROME VALUE 0~29 REGISTER (NFWSYNDRONE0 ~ NFWSYNDRONE29)				
Address : 0XC005_12F4 ~ 0XC005_1368				
[31:30]	R	RESERVED	Reserved	2'b0
[29:16]	W	SYNDROM[3+(NX4)]	ECC Decoder Result Odd Syndrome Data3(offset n x 4)	14'h0
[15:14]	R	RESERVED	Reserved	2'b0
[13:0]	W	SYNDROM[1+(NX4)]	ECC Decoder Result Odd Syndrome Data1 (offset n x 4)	14'h0
NAND FLASH DATA REGISTER (NFDATA)				
Address : Shadow 0 : 0x2C00_0000				
[31:0]	R/W	NFDATA	Nand flash data register. In case of 16 bit access on this register, it generates 8 bit access cycle in twice automatically. In case of 32 bit access on this register, it also generates four 8 bit access cycles automatically.	32'hx
NAND FLASH COMMAND REGISTER (NFCMD)				
Address : Shadow 0 : 0x 2C00_0010				
[15:8]	-	RESERVED	Reserved	8'hx
[7:0]	W	NFCMD	Nand flash command register. Writing on this register generates a command cycle with CLE signal and transfers this value on data bus automatically. You have to write only 8 bit data on this register. Do not access this register with 16/32 bit data.	8'hx
NAND FLASH ADDRESS REGISTER (NFADDR)				
Address : Shadow 0 : 0x 2C00_0018				
[15:8]	-	RESERVED	Reserved	8'hx
[7:0]	W	NFADDR	Nand flash address register. Writing on this register generates an address cycle with ALE signal and transfers this value on data bus automatically. You have to write only 8 bit data on this register. Do not access this register with 16/32 bit access. Only byte access is available.	8'hx

Section 15. *GPIO Controller*

15.1 Overview

15.1.1 Features

- Programmable Pull-Up Control
- Edge/Level Detect
- Supports programmable Pull-Up resistance.
- Supports four event detection modes
 - Rising Edge Detection
 - Falling Edge Detection
 - Low Level Detection
 - High Level Detection
- The number of GPIOs : 160

15.1.2 Block Diagram

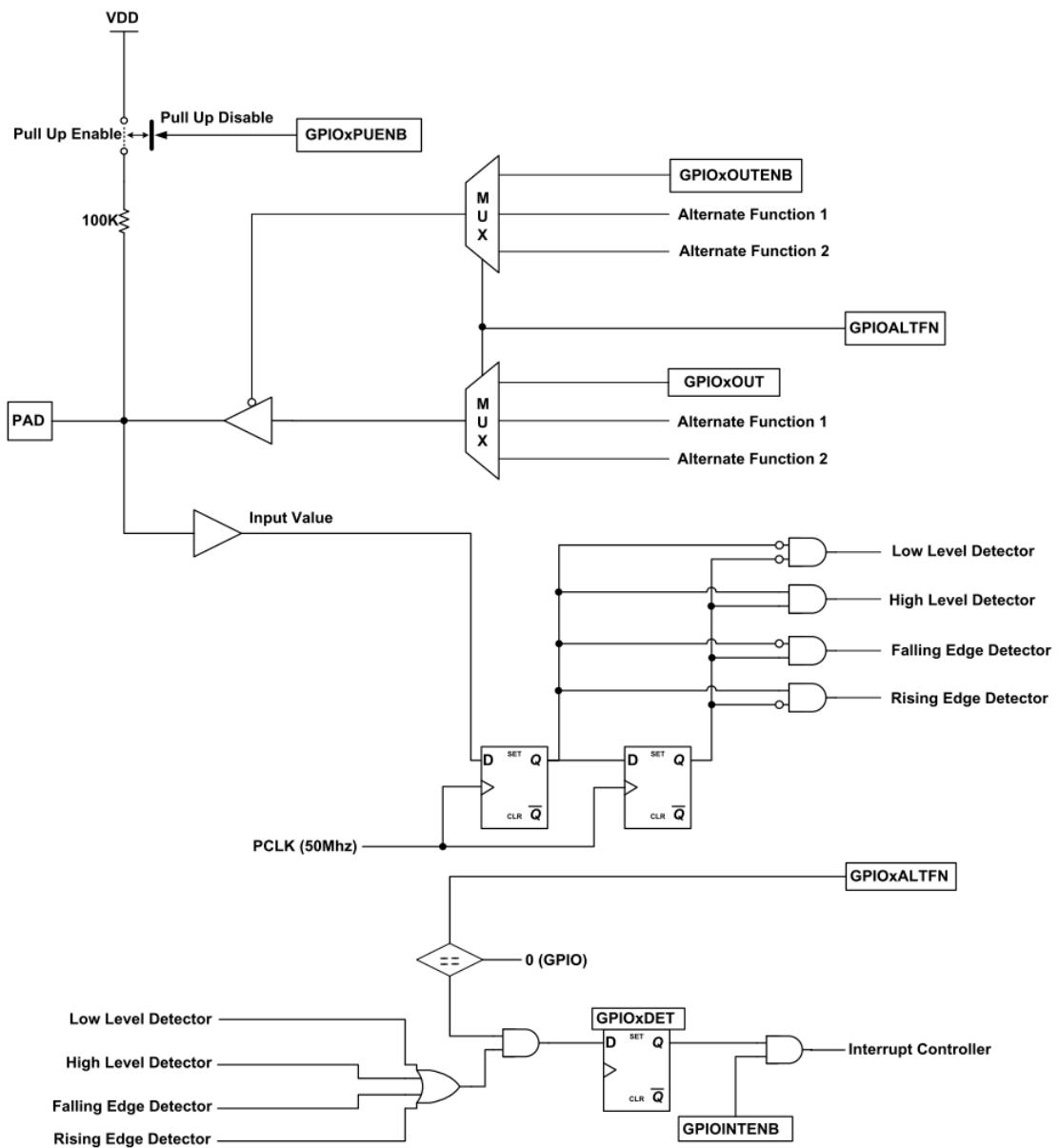


Figure 15-1. GPIO Block Diagram

15.2 Functional Description

NXP4330D/Q GPIO Pins have an internal pull-up resistance of 100K ohm.

The current of the pull-up resistance (for VDD = 3.3 V and V (PAD) = 0V) is listed in the table below:

Pull Up	MIN	TYP	MAX	UNIT
ENABLE	10	33	72	uA
DISABLE	-	-	0.1	uA

Table 15-1. Pull-Up Resister Current

Most NXP4330D/Q GPIO ports contain the Alternate function (some ports supports up to Alternate Function2). All GPIO ports should be set as GPIO function or Alternate Function suitable for the user's purposes and this setting can easily be performed with GPIO registers. In addition, all GPIO pull-up resistances are enabled/ disabled. This setting operates when the system is fully booted and does not affect the system in initial booting. If there is a value which needs to be determined in system booting, the value is given by inserting a pull up/down resistance from outside.

15.2.1 Input Operation

To use GPIO for input, the GPIO function should be selected by setting the relevant bit of the GPIO Alternate Function Select register as b'00 to select the GPIO function. In addition, the GPIO Input mode should be, also, selected by the GPIOx Output Enable register (*GPIOxOUTENB*) as '0'.

An input signal is detected by selecting a desired detection type with the GPIOx Event Detect Mode register. Four types of input signal can be detected: Low Level, High Level, Falling edge and Rising edge. The GPIOx Event Detect Mode registers consist of GPIOx Event Detect Mode register0 (*GPIOxDETMODE0*) and GPIOx Event Detect Mode register1 (*GPIOxDETMODE1*).

To use interrupt, set the GPIOx Interrupt Enable Register (*GPIOxINTENB*) as '1'.

The GPIOx Event Detect Register (*GPIOxDET*) enables checking the generation of an event via GPIO and may be used as the Pending Clear function when an interrupt occurs.

When the GPIOx PAD Status Register (*GPIOxPAD*) is set as GPIO Input mode, the level of the relevant GPIOx PAD can be checked.

The Open drain pins (GPIOB[7:4] and GPIOC[8]) operate in Input mode only when the GPIOx Output register (*GPIOxOUT*) is set as '1'. The Open drain pins are operated by the GPIOx Output Register (*GPIOxOUT*) even if the GPIOx Output Enable register (*GPIOxOUTENB*) is set as Input mode.

15.2.2 Output Operation

To use GPIO for output, the GPIO function should be selected by setting the relevant bit of the GPIOx. Alternate Function Select register should be set as b'00 to select the GPIO function. In addition, the GPIOx Output mode should also be selected by setting the GPIOx Output Enable register as '1'.

If you set a desired output value (low level: '0', high level: '1') with the GPIOx Output Register (*GPIOxOUT*), the value is reflected to the corresponding bit.

The Open drain pins (GPIOB[7:4] and GPIOC[8]) operate in output mode only when the GPIOx Output register (*GPIOxOUT*) is set as '0'. The Open drain pins are operated by the GPIOx Output Register (*GPIOxOUT*) even if the GPIOx Output Enable register (*GPIOxOUTENB*) is set as Input mode.

15.2.3 Alternate Function Operation

Among the 151 GPIO pins of the NXP4330D/Q , most GPIO pins have an Alternate function. However, the Alternate function and the GPIO function should not be used simultaneously. Therefore, Alternate Function1 and Alternate Function2 are operated by setting the corresponding bits of the GPIOx Alternate Function Select register as b'01 and b'10, respectively.

15.3 Register Summary

Note : GPIOx Register (x = A, B, C, D, E)

Bit	R/W	Symbol	Description	Reset Value
[15:14]	R/W	GPIOXDETMODE0_7	GPIOx[7]: Specifies Detect mode when GPIOx 7pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[13:12]	R/W	GPIOXDETMODE0_6	GPIOx[6]: Specifies Detect mode when GPIOx 6pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[11:10]	R/W	GPIOXDETMODE0_5	GPIOx[5]: Specifies Detect mode when GPIOx 5pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[9:8]	R/W	GPIOXDETMODE0_4	GPIOx[4]: Specifies Detect mode when GPIOx 4pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[7:6]	R/W	GPIOXDETMODE0_3	GPIOx[3]: Specifies Detect mode when GPIOx 3pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[5:4]	R/W	GPIOXDETMODE0_2	GPIOx[2]: Specifies Detect mode when GPIOx 2pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[3:2]	R/W	GPIOXDETMODE0_1	GPIOx[1]: Specifies Detect mode when GPIOx 1pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[1:0]	R/W	GPIOXDETMODE0_0	GPIOx[0]: Specifies Detect mode when GPIOx 0pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
GPIOx EVENT DETECT MODE REGISTER 1 (GPIOxDETMODE1)				
<i>Address : GPIOA: C001A00Ch / GPIOB: C001B00Ch / GPIOC: C001C00Ch / GPIOD: C001D00Ch / GPIOE: C001E00Ch</i>				
[31:30]	R/W	GPIOXDETMODE1_31	GPIOx[31]: Specifies Detect mode when GPIOx 31pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[29:28]	R/W	GPIOXDETMODE1_30	GPIOx[30]: Specifies Detect mode when GPIOx 30pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[27:26]	R/W	GPIOXDETMODE1_29	GPIOx[29]: Specifies Detect mode when GPIOx 29pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[25:24]	R/W	GPIOXDETMODE1_28	GPIOx[28]: Specifies Detect mode when GPIOx 28pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[23:22]	R/W	GPIOXDETMODE1_27	GPIOx[27]: Specifies Detect mode when GPIOx 27pin is in Input mode. 00 : Low Level 01 : High Level 10 : Falling Edge 11 : Rising Edge	2b00
[21:20]	R/W	GPIOXDETMODE1_26	GPIOx[26]: Specifies Detect mode when GPIOx 26pin is in Input mode.	2b00

Bit	R/W	Symbol	Description	Reset Value
			00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	
[19:18]	R/W	GPIOXDETMODE1_25	GPIOx[25]: Specifies Detect mode when GPIOx 25pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[17:16]	R/W	GPIOXDETMODE1_24	GPIOx[24]: Specifies Detect mode when GPIOx 24pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[15:14]	R/W	GPIOXDETMODE1_23	GPIOx[23]: Specifies Detect mode when GPIOx 23pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[13:12]	R/W	GPIOXDETMODE1_22	GPIOx[22]: Specifies Detect mode when GPIOx 22pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[11:10]	R/W	GPIOXDETMODE1_21	GPIOx[21]: Specifies Detect mode when GPIOx 21pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[9:8]	R/W	GPIOXDETMODE1_20	GPIOx[20]: Specifies Detect mode when GPIOx 20pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[7:6]	R/W	GPIOXDETMODE1_19	GPIOx[19]: Specifies Detect mode when GPIOx 19pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[5:4]	R/W	GPIOXDETMODE1_18	GPIOx[18]: Specifies Detect mode when GPIOx 18pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[3:2]	R/W	GPIOXDETMODE1_17	GPIOx[17]: Specifies Detect mode when GPIOx 17pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
[1:0]	R/W	GPIOXDETMODE1_16	GPIOx[16]: Specifies Detect mode when GPIOx 16pin is in Input mode. 00 : Low Level 10 : Falling Edge 01 : High Level 11 : Rising Edge	2b00
GPIOx INTERRUPT ENABLE REGISTER (GPIOxINTENB)				
Address : GPIOA: C001A010h / GPIOB: C001B010h / GPIOC: C001C010h / GPIOD: C001D010h / GPIOE: C001E010h				
[31:0]	R/W	GPIOXINTENB	GPIOx[31:0]: Specifies the use of an interrupt when a GPIOx Event occurs. The events specified in GPIOxDETMODE0 and GPIOxDETMODE1 are used. 0 : Disable 1 : Enable	32'h0
GPIOx EVENT DETECT REGISTER (GPIOxDET)				
Address : GPIOA: C001A014h / GPIOB: C001B014h / GPIOC: C001C014h / GPIOD: C001D014h / GPIOE: C001E014h				
[31:0]	R/W	GPIOXDET	GPIOx[31:0]: Shows if an event is detected in accordance with Event Detect mode in GPIOx Input Mode. Set '1' to clear the relevant bit. GPIOx[31:0] is used as a Pending register when an interrupt occurs. Read : 0 : Not Detect 1 : Detected	32'h0

Bit	R/W	Symbol	Description	Reset Value
			Write : 0: Not Clear 1: Clear	
GPIOx PAD STATUS REGISTER (GPIOxPAD)				
<i>Address : GPIOA: C001A018h / GPIOB: C001B018h / GPIOC: C001C018h / GPIOD: C001D018h / GPIOE: C001E018h</i>				
[31:0]	R	GPIOXPAD	GPIOx[31:0]: Can read the level pf PAD in GPIOx Input mode. The data read in this register is the data not passing a filter and reflects the PAD status itself. 0 : Low Level 1 : High Level	32'h0
Reserved				
<i>Address : GPIOA: C001A01Ch / GPIOB: C001B01Ch / GPIOC: C001C01Ch / GPIOD: C001D01Ch / GPIOE: C001E01Ch</i>				
[31:15]	-	RESERVED	Reserved	-
GPIOx ALTERNATE FUNCTION SELECT REGISTER 0 (GPIOxALTFN0)				
<i>Address : GPIOA: C001A020h / GPIOB: C001B020h / GPIOC: C001C020h / GPIOD: C001D020h / GPIOE: C001E020h</i>				
[31:30]	R/W	GPIOXALTFN0_15	GPIOx[15]: Selects the function of GPIOx 15pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[29:28]	R/W	GPIOXALTFN0_14	GPIOx[14]: Selects the function of GPIOx 14pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[27:26]	R/W	GPIOXALTFN0_13	GPIOx[13]: Selects the function of GPIOx 13pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[25:24]	R/W	GPIOXALTFN0_12	GPIOx[12]: Selects the function of GPIOx 12pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[23:22]	R/W	GPIOXALTFN0_11	GPIOx[11]: Selects the function of GPIOx 11pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[21:20]	R/W	GPIOXALTFN0_10	GPIOx[10]: Selects the function of GPIOx 10pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[19:18]	R/W	GPIOXALTFN0_9	GPIOx[9]: Selects the function of GPIOx 9pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[17:16]	R/W	GPIOXALTFN0_8	GPIOx[8]: Selects the function of GPIOx 8pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[15:14]	R/W	GPIOXALTFN0_7	GPIOx[7]: Selects the function of GPIOx 7pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[13:12]	R/W	GPIOXALTFN0_6	GPIOx[6]: Selects the function of GPIOx 6pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[11:10]	R/W	GPIOXALTFN0_5	GPIOx[5]: Selects the function of GPIOx 5pin.	2'b00

Bit	R/W	Symbol	Description	Reset Value
			00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	
[9:8]	R/W	GPIOXALTFN0_4	GPIOx[4]: Selects the function of GPIOx 4pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[7:6]	R/W	GPIOXALTFN0_3	GPIOx[3]: Selects the function of GPIOx 3pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[5:4]	R/W	GPIOXALTFN0_2	GPIOx[2]: Selects the function of GPIOx 2pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[3:2]	R/W	GPIOXALTFN0_1	GPIOx[1]: Selects the function of GPIOx 1pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[1:0]	R/W	GPIOXALTFN0_0	GPIOx[0]: Selects the function of GPIOx 0pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00

GPIOx ALTERNATE FUNCTION SELECT REGISTER 1 (GPIOXALTFN1)

Address : GPIOA: C001A024h / GPIOB: C001B024h / GPIOC: C001C024h / GPIOD: C001D024h / GPIOE: C001E024h

[31:30]	R/W	GPIOXALTFN1_31	GPIOx[31]: Selects the function of GPIOx 31pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[29:28]	R/W	GPIOXALTFN1_30	GPIOx[30]: Selects the function of GPIOx 30pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[27:26]	R/W	GPIOXALTFN1_29	GPIOx[29]: Selects the function of GPIOx 29pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[25:24]	R/W	GPIOXALTFN1_28	GPIOx[28]: Selects the function of GPIOx 28pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[23:22]	R/W	GPIOXALTFN1_27	GPIOx[27]: Selects the function of GPIOx 27pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[21:20]	R/W	GPIOXALTFN1_26	GPIOx[26]: Selects the function of GPIOx 26pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[19:18]	R/W	GPIOXALTFN1_25	GPIOx[25]: Selects the function of GPIOx 25pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00
[17:16]	R/W	GPIOXALTFN1_24	GPIOx[24]: Selects the function of GPIOx 24pin. 00 : ALT Function0 10 : ALT Function2 11 : ALT Function3	2b00

Bit	R/W	Symbol	Description	Reset Value
[15:14]	R/W	GPIOXALTFN1_23	GPIOx[23]: Selects the function of GPIOx 23pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[13:12]	R/W	GPIOXALTFN1_22	GPIOx[22]: Selects the function of GPIOx 22pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[11:10]	R/W	GPIOXALTFN1_21	GPIOx[21]: Selects the function of GPIOx 21pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[9:8]	R/W	GPIOXALTFN1_20	GPIOx[20]: Selects the function of GPIOx 20pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[7:6]	R/W	GPIOXALTFN1_19	GPIOx[19]: Selects the function of GPIOx 19pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[5:4]	R/W	GPIOXALTFN1_18	GPIOx[18]: Selects the function of GPIOx 18pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[3:2]	R/W	GPIOXALTFN1_17	GPIOx[17]: Selects the function of GPIOx 17pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
[1:0]	R/W	GPIOXALTFN1_16	GPIOx[16]: Selects the function of GPIOx 16pin. 00 : ALT Function0 01 : ALT Function1 10 : ALT Function2 11 : ALT Function3	2'b00
GPIOx DETECT ENABLE REGISTER(GPIOxDETENB)				
<i>Address : GPIOA: C001A03Ch / GPIOB: C001B03Ch / GPIOC: C001C03Ch / GPIOD: C001D03Ch / GPIOE: C001E03Ch</i>				
[31:0]	R/W	GPIOXDETENB	GPIOx[31:0]: Decides the use of the detected mode of GPIOx PAD. 0: Disable 1: Enable	32'h0
GPIOx SLEW REGISTER(GPIOx_SLEW)				
<i>Address : GPIOA: C001A040h / GPIOB: C001B040h / GPIOC: C001C040h / GPIOD: C001D040h / GPIOE: C001E040h</i>				
[31:0]	R/W	GPIOX_SLEW	GPIOx[31:0]: Decides the use of the Slew resistance of GPIOx PAD. 0: Fast Slew 1: Normal Slew	32'hfffffff
GPIOx SLEW DISABLE DEFAULT REGISTER(GPIOx_SLEW_DISABLE_DEFAULT)				
<i>Address : GPIOA: C001A044h / GPIOB: C001B044h / GPIOC: C001C044h / GPIOD: C001D044h / GPIOE: C001E044h</i>				
[31:0]	R/W	GPIOX_SLEW_DISABLE_DEFault	GPIOx[31:0]: Decides the use of the Slew resistance of GPIOx PAD. 0: Use Default Slew 1: Use GPIOx_Slew Value	32'hfffffff
GPIOx DRV1 REGISTER(GPIOx_DRV1)				
<i>Address : GPIOA: C001A048h / GPIOB: C001B048h / GPIOC: C001C048h / GPIOD: C001D048h / GPIOE: C001E048h</i>				
[31:0]	R/W	GPIOX_DRV1	GPIOx[31:0]: Decides the use of the Drive Strength1 resistance of GPIOx PAD. 0: Drive Strength0 to 0 1: Drive Strength0 to 1	32'h0
GPIOx DRV1 DISABLE DEFAULT REGISTER(GPIOx_DRV1_DISABLE_DEFAULT)				
<i>Address : GPIOA: C001A04Ch / GPIOB: C001B04Ch / GPIOC: C001C04Ch / GPIOD: C001D04Ch / GPIOE: C001E04Ch</i>				

Bit	R/W	Symbol	Description	Reset Value			
[31:0]	R/W	GPIOx_DRV1_DISABLE_DEF AULT	GPIOx[31:0]: Decides the use of the Drive Strength1 resistance of GPIOx PAD. 0: Use Default Drive Strength 1: Use GPIOx_DRV1 Value	32hfffffff			
GPIOx DRV0 REGISTER(GPIOx_DRV0)							
Address : GPIOA: C001A050h / GPIOB: C001B050h / GPIOC: C001C050h / GPIOD: C001D050h / GPIOE: C001E050h							
[31:0]	R/W	GPIOx_DRV0	GPIOx[31:0]: Decides the use of the Drive Strength0 resistance of GPIOx PAD. 0: Drive Strength1 to 0 1: Drive Strength1 to 1 DC current of output driver	32'h0			
			VDD=1.8V				
			2.5mA				
			2.6mA				
			DS0=0, DS1=0				
			3.8mA				
(DS : Drive Strength)							
GPIOx DRV0 DISABLE DEFAULT REGISTER(GPIOx_DRV0_DISABLE_DEFAULT)							
Address : GPIOA: C001A054h / GPIOB: C001B054h / GPIOC: C001C054h / GPIOD: C001D054h / GPIOE: C001E054h							
[31:0]	R/W	GPIOx_DRV0_DISABLE_DEF AULT	GPIOx[31:0]: Decides the use of the Drive Strength0 resistance of GPIOx PAD. 0: Use Default Drive Strength 1: Use GPIOx_DRV0 Value	32hfffffff			
GPIOx PULLSEL REGISTER(GPIOx_PULLSEL)							
Address : GPIOA: C001A058h / GPIOB: C001B058h / GPIOC: C001C058h / GPIOD: C001D058h / GPIOE: C001E058h							
[31:0]	R/W	GPIOx_PULLSEL	GPIOx[31:0]: Decides the Pull-up or Pull-down of GPIOx PAD. 0: Pull-Down 1: Pull-Up	32'h0			
GPIOx PULLSEL DISABLE DEFAULT REGISTER(GPIOx_PULLSEL_DISABLE_DEFAULT)							
Address : GPIOA: C001A05Ch / GPIOB: C001B05Ch / GPIOC: C001C05Ch / GPIOD: C001D05Ch / GPIOE: C001E05Ch							
[31:0]	R/W	GPIOx_PULLSEL_DISABLE_DEFAULT	GPIOx[31:0]: Decides the use of the PullSel resistance of GPIOx PAD. 0: Use Default Pull Sel 1: Use GPIOx_PULLSEL Value	32'h0			
GPIOx PULLENB REGISTER(GPIOx_PULLENB)							
Address : GPIOA: C001A060h / GPIOB: C001B060h / GPIOC: C001C060h / GPIOD: C001D060h / GPIOE: C001E060h							
[31:0]	R/W	GPIOx_PULLENB	GPIOx[31:0]: Decides the use of the Pull-up resistance (100 Kohm) of GPIOx PAD. 0: Pull-Up Disable 1: Pull-Up Enable	32'h0			
GPIOx PULLENB DISABLE DEFAULT REGISTER(GPIOx_PULLENB_DISABLE_DEFAULT)							
Address : GPIOA: C001A064h / GPIOB: C001B064h / GPIOC: C001C064h / GPIOD: C001D064h / GPIOE: C001E064h							
[31:0]	R/W	GPIOx_PULLENB_DISABLE_DEFAULT	GPIOx[31:0]: Decides the use of the PullEnb resistance of GPIOx PAD. 0: Use Default Pull Enb 1: Use GPIOx_PULLENB Value	32'h0			

Section 16. Ethernet MAC

16.1 Overview

The GMAC enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2008 standard.

EMAC supports 10/100/1000Mbps data transfer rates, and it has Reduced Gigabit Media Independent Interface (RGMII) with External PHY chip.

16.1.1 MAC Core features

- Supports 10/100/1000 Mbps data transfer rates with the following phy interfaces:
 - RGMII interface to communicate with an external gigabit PHY
- Supports both full-duplex and half-duplex operation
- Automatic CRC and pad generation controllable on receive frames
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16KB
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Support a variety of flexible address filtering mode
- Separate transmission, reception, and control interface to the Application
- MDIO Master Interface for PHY device configuration and management
- Compliant to the following standards:
 - IEEE 802.3 – 2002 for Ethernet MAC
 - IEEE 1588 – 2002 standard for precision networked clock synchronization
 - RGMII specification version 2.0 from HP/Marvell.

16.1.2 DMA block features

- 32/64/128-bit data transfer
- Single-channel Transmit and Receive engines
- Optimization for packet-oriented DMA transfers with frame delimiters
- Byte-aligned addressing for data buffer support
- Descriptor architectures, allowing large blocks of data transfer with minimum CPU intervention; each descriptor can transfer up to 8KB of data
- Individual programmable burst size for Transmit and Receive DMA Engines for optimal host bus utilization
- Programmable interrupt options for different operational conditions
- Per-frame Transmit/Receive complete interrupt control
- Round-robin or fixed-priority arbitration between Receive and Transmit Engines
- Start/Stop mode
- Separate port for host CSR (Control and Status Register) access and host data interface

16.1.3 Block Diagram

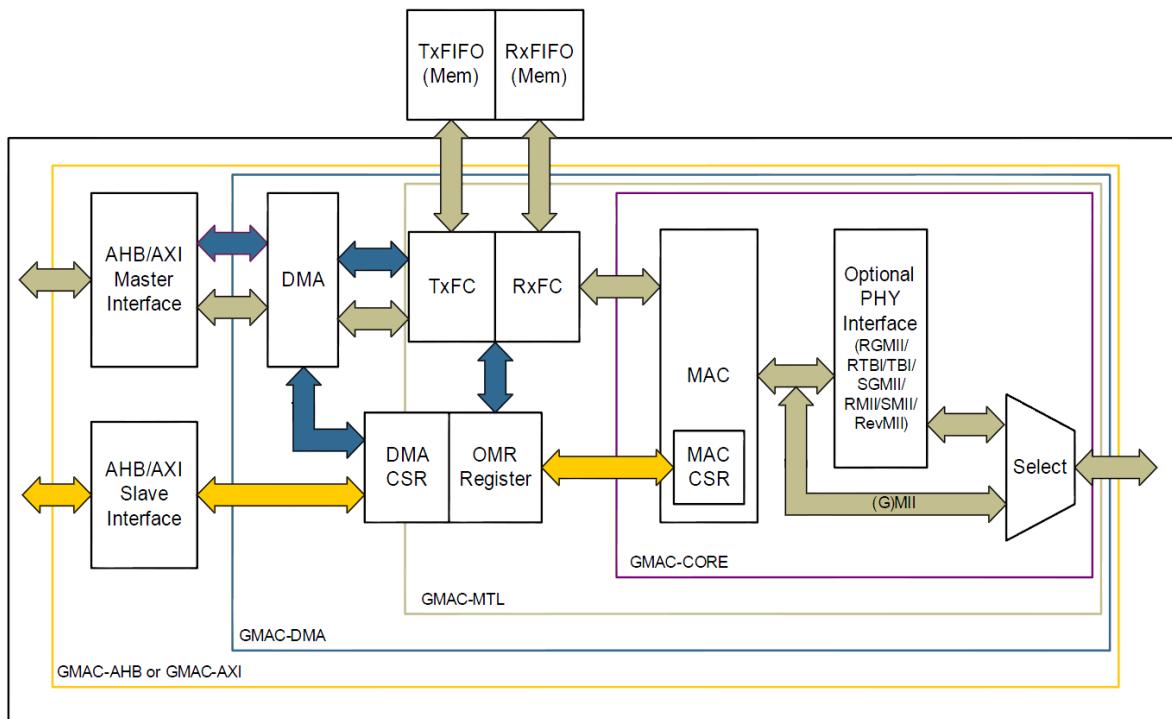


Figure 16-1 Block Diagram of the Ethernet MAC

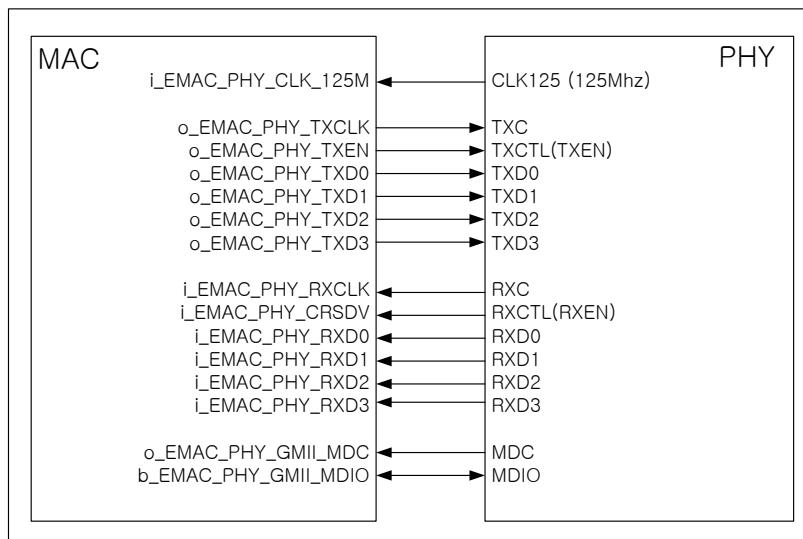


Figure 16-2 RGMII Interface between MAC and Gigabit Ethernet PHY

16.2 Register Summary

16.2.1 MAC DMA Register Description

Bit	R/W	Symbol	Description	Reset Value				
Ethernet MAC DMA Register 0 (Bus Mode Register)								
Address : C006_1000h								
[31]	R/W	RIB	<p>Rebuild INCRx Burst When this bit is set high and the AHB master gets an EBT (Retry, Split, or Losing bus grant), the AHB master interface rebuilds the pending beats of any burst transfer initiated with INCRx. The AHB master interface rebuilds the beats with a combination of specified bursts with INCRx and SINGLE. By default, the AHB master interface rebuilds pending beats of an EBT with an unspecified (INCR) burst.</p> <p>This bit is valid only in the GMAC-AHB configuration. It is reserved in all other configuration.</p>	1'b0				
[30]	-	RESERVED	Reserved	1'b0				
[29:28]	R/W	PRWG	<p>Channel Priority Weights This field sets the priority weights for Channel 0 during the round-robin arbitration between the DMA channels for the system bus.</p> <table> <tr> <td>00: The priority weight is 1.</td> <td>01: The priority weight is 2.</td> </tr> <tr> <td>10: The priority weight is 3.</td> <td>11: The priority weight is 4.</td> </tr> </table> <p>This field is present in all DWC_gmac configurations except GMAC-AXI when you select the AV feature. Otherwise, this field is reserved and read-only (RO).</p>	00: The priority weight is 1.	01: The priority weight is 2.	10: The priority weight is 3.	11: The priority weight is 4.	2'b0
00: The priority weight is 1.	01: The priority weight is 2.							
10: The priority weight is 3.	11: The priority weight is 4.							
[27]	R/W	TXPR	<p>Transmit Priority When set, this bit indicates that the transmit DMA has higher priority than the receive DMA during arbitration for the system-side bus. In the GMAC-AXI configuration, this bit is reserved and read-only (RO).</p>	1'b0				
[26]	R/W	MB	<p>Mixed Burst When this bit is set high and the FB bit is low, the AHB master interface starts all bursts of length more than 16 with INCR (undefined burst), whereas it reverts to fixed burst transfers (INCRx and SINGLE) for burst length of 16 and less.</p>	1'b0				
[25]	R/W	AAL	<p>Address-Aligned Beats When this bit is set high and the FB bit is equal to 1, the AHB or AXI interface generates all bursts aligned to the start address LS bits. If the FB bit is equal to 0, the first burst (accessing the start address of data buffer) is not aligned, but subsequent bursts are aligned to the address.</p>	1'b0				
[24]	R/W	PBLX8	<p>PBLx8 Mode When set high, this bit multiplies the programmed PBL value (Bits [22:17] and Bits[13:8]) eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.</p>	1'b0				
[23]	R/W	USP	<p>Use Separate PBL When set high, this bit configures the Rx DMA to use the value configured in Bits [22:17] as PBL. The PBL value in Bits [13:8] is applicable only to the Tx DMA operations.</p> <p>When reset to low, the PBL value in Bits [13:8] is applicable for both DMA engines.</p>	1'b0				
[22:17]	R/W	RPBL	<p>Rx DMA PBL This field indicates the maximum number of beats to be transferred in one Rx DMA transaction. This is the maximum value that is used in a single block Read or Write.</p> <p>The Rx DMA always attempts to burst as specified in the RPBL bit each time it starts a Burst transfer on the host bus. You can program RPBL with values of 1, 2, 4, 8, 16, and 32. Any other value results in undefined behavior.</p> <p>This field is valid and applicable only when USP is set high.</p>	6'h10				

Bit	R/W	Symbol	Description	Reset Value
[16]	R/W	FB	<p>Fixed Burst</p> <p>This bit controls whether the AHB or AXI master interface performs fixed burst transfers or not. When set, the AHB interface uses only SINGLE, INCR4, INCR8, or INCR16 during start of the normal burst transfers. When reset, the AHB or AXI interface uses SINGLE and INCR burst transfer operations.</p> <p>For more information, see Bit 0 (UNDEF) of the AXI Bus Mode register in the GMAC-AXI configuration.</p>	1'b0
[15:14]	R/W	PR	<p>Priority Ratio</p> <p>These bits control the priority ratio in the weighted round-robin arbitration between the Rx DMA and Tx DMA. These bits are valid only when Bit 1 (DA) is reset. The priority ratio is Rx:Tx or Tx:Rx depending on whether Bit 27 (TXPR) is reset or set.</p> <p>00: The Priority Ratio is 1:1. 01: The Priority Ratio is 2:1. 10: The Priority Ratio is 3:1. 11: The Priority Ratio is 4:1.</p> <p>In the GMAC-AXI configuration, these bits are reserved and read-only (RO).</p>	2'b0
[13:8]	R/W	PBL	<p>Programmable Burst Length</p> <p>These bits indicate the maximum number of beats to be transferred in one DMA transaction. This is the maximum value that is used in a single block Read or Write. The DMA always attempts to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value results in undefined behavior. When USP is set high, this PBL value is applicable only for Tx DMA transactions.</p> <p>If the number of beats to be transferred is more than 32, then perform the following steps:</p> <ol style="list-style-type: none"> 1. Set the PBLx8 mode. 2. Set the PBL. <p>For example, if the maximum number of beats to be transferred is 64, then first set PBLx8 to 1 and then set PBL to 8. The PBL values have the following limitation: The maximum number of possible beats (PBL) is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified.</p>	6'h01
[7]	R/W	ATDS	<p>Alternate Descriptor Size</p> <p>When set, the size of the alternate descriptor increases to 32 bytes (8 DWORDS). This is required when the Advanced Timestamp feature or the IPC Full Checksum Offload Engine (Type 2) is enabled in the receiver. The enhanced descriptor is not required if the Advanced Timestamp and IPC Full Checksum Offload Engine (Type 2) features are not enabled. In such case, you can use the 16 bytes descriptor to save 4 bytes of memory.</p> <p>This bit is present only when you select the Alternate Descriptor feature and any one of the following features during core configuration:</p> <ul style="list-style-type: none"> ■ Advanced Timestamp feature ■ IPC Full Checksum Offload Engine (Type 2) feature <p>Otherwise, this bit is reserved and is read-only.</p> <p>When reset, the descriptor size reverts back to 4 DWORDs (16 bytes).</p> <p>This bit preserves the backward compatibility for the descriptor size. In versions prior to 3.50a, the descriptor size is 16 bytes for both normal and enhanced descriptors. In version 3.50a, descriptor size is increased to 32 bytes because of the Advanced Timestamp and IPC Full Checksum Offload Engine (Type 2) features.</p>	1'b0
[6:2]	R/W	DSL	Descriptor Skip Length	5'b0
[1]	R/W	DA	DMA Arbitration Scheme	1'b0
			This bit specifies the arbitration scheme between the transmit and receive paths of Channel 0.	
			0: Weighted round-robin with Rx:Tx or Tx:Rx - The priority between the paths is according to the	

Bit	R/W	Symbol	Description	Reset Value
			<p>priority specified in Bits [15:14] (PR) and priority weights specified in Bit 27 (TXPR).</p> <p>1: Fixed priority - The transmit path has priority over receive path when Bit 27 (TXPR) is set. Otherwise, receive path has priority over the transmit path.</p> <p>In the GMAC-AXI configuration, these bits are reserved and are read-only (RO).</p>	
[0]	R/W	SWR	<p>Software Reset</p> <p>When this bit is set, the MAC DMA Controller resets the logic and all internal registers of the MAC. It is cleared automatically after the reset operation is complete in all of the DWC_gmac clock domains. Before reprogramming any register of the DWC_gmac, you should read a zero (0) value in this bit.</p> <p>Note:</p> <ul style="list-style-type: none"> ■ The Software reset function is driven only by this bit. Bit 0 of Register 64 (Channel 1 Bus Mode Register) or Register 128 (Channel 2 Bus Mode Register) has no impact on the Software reset function. ■ The reset operation is completed only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for the software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock. 	1'b1
Ethernet MAC DMA Register 1 (Transmit Poll Demand Register)				
Address : C006_1004h				
[31:0]	R/W	TPD	<p>Transmit Poll Demand</p> <p>When these bits are written with any value, the DMA reads the current descriptor to which the Register 18 (Current Host Transmit Descriptor Register) is pointing. If that descriptor is not available (owned by the Host), the transmission returns to the Suspend state and Bit 2 (TU) of Register 5 (Status Register) is asserted. If the descriptor is available, the transmission resumes.</p> <p>When this register is read, it always returns zero.</p>	32'b0
Ethernet MAC DMA Register 2 (Receive Poll Demand Register)				
Address : C006_1008h				
[31:0]	R/W	RPD	<p>Receive Poll Demand</p> <p>When these bits are written with any value, the DMA reads the current descriptor to which the Register 19 (Current Host Receive Descriptor Register) is pointing. If that descriptor is not available (owned by the Host), the reception returns to the Suspended state and Bit 7 (RU) of Register 5 (Status Register) is asserted. If the descriptor is available, the Rx DMA returns to the active state.</p> <p>When this register is read, it always returns zero.</p>	32'b0
Ethernet MAC DMA Register 3 (Receive Descriptor List Address Register)				
Address : C006_100Ch				
[31:0]	R/W	RDESLA	<p>Start of Receive List</p> <p>This field contains the base address of the first descriptor in the Receive Descriptor list. The LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width are ignored and internally taken as all-zero by the DMA. Therefore, these LSB bits are read-only (RO).</p>	32'b0
Ethernet MAC DMA Register 4 (Transmit Descriptor List Address Register)				
Address : C006_10010h				
[31:0]	R/W	TDESLA	<p>Start of Transmit List</p> <p>This field contains the base address of the first descriptor in the Transmit Descriptor list. The LSB bits (1:0, 2:0, 3:0) for 32-bit, 64-bit, or 128-bit bus width are ignored and are internally taken as all-zero by the DMA. Therefore, these LSB bits are read-only (RO).</p>	32'b0
Ethernet MAC DMA Register 5 (Status Register)				
Address : C006_1014h				
[31]	-	RESERVED	Reserved	1'b0

Bit	R/W	Symbol	Description	Reset Value
[30]	R	GLPII / GTMSI	<p>GLPII: GMAC LPI Interrupt (for Channel 0)</p> <p>This bit indicates an interrupt event in the LPI logic of the MAC. To reset this bit to 1'b0, the software must read the corresponding registers in the DWC_gmac to get the exact cause of the interrupt and clear its source.</p> <p>Note:</p> <p>GLPII status is given only in Channel 0 DMA register and is applicable only when the Energy Efficient Ethernet feature is enabled. Otherwise, this bit is reserved. When this bit is high, the interrupt signal from the MAC (sbd_intr_o) is high.</p> <p>-or-</p> <p>GTMSI: GMAC TMS Interrupt (for Channel 1 and Channel 2)</p> <p>This bit indicates an interrupt event in the traffic manager and scheduler logic of DWC_gmac. To reset this bit, the software must read the corresponding registers (Channel Status Register) to get the exact cause of the interrupt and clear its source.</p> <p>Note:</p> <p>GTMSI status is given only in Channel 1 and Channel 2 DMA register when the AV feature is enabled and corresponding additional transmit channels are present. Otherwise, this bit is reserved. When this bit is high, the interrupt signal from the MAC (sbd_intr_o) is high.</p>	1'b0
[29]	R	TTI	<p>Timestamp Trigger Interrupt</p> <p>This bit indicates an interrupt event in the Timestamp Generator block of the DWC_gmac. The software must read the corresponding registers in the DWC_gmac to get the exact cause of the interrupt and clear its source to reset this bit to 1'b0. When this bit is high, the interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high.</p> <p>This bit is applicable only when the IEEE 1588 Timestamp feature is enabled. Otherwise, this bit is reserved.</p>	1'b0
[28]	R	GPI	<p>GMAC PMT Interrupt</p> <p>This bit indicates an interrupt event in the PMT module of the DWC_gmac. The software must read the PMT Control and Status Register in the MAC to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high when this bit is high.</p> <p>This bit is applicable only when the Power Management feature is enabled. Otherwise, this bit is reserved.</p> <p>Note: The GPI and pmt_intr_o interrupts are generated in different clock domains.</p>	1'b0
[27]	R	GMI	<p>GMAC MMC Interrupt</p> <p>This bit reflects an interrupt event in the MMC module of the DWC_gmac. The software must read the corresponding registers in the DWC_gmac to get the exact cause of the interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high when this bit is high.</p> <p>This bit is applicable only when the MAC Management Counters (MMC) are enabled. Otherwise, this bit is reserved.</p>	1'b0
[26]	R	GLI	<p>GMAC Line Interface Interrupt</p> <p>When set, this bit reflects any of the following interrupt events in the DWC_gmac interfaces (if present and enabled in your configuration):</p> <ul style="list-style-type: none"> ■ PCS (TBI, RTBI, or SGMII): Link change or auto-negotiation complete event ■ SMII or RGMI: Link change event ■ General Purpose Input Status (GPIS): Any LL or LH event on the gpi_i input ports <p>To identify the exact cause of the interrupt, the software must first read Bit 11 and Bits[2:0] of Register 14 (Interrupt Status Register) and then to clear the source of interrupt (which also clears the GLI interrupt), read any of the following corresponding registers:</p> <ul style="list-style-type: none"> ■ PCS (TBI, RTBI, or SGMII): Register 49 (AN Status Register) ■ SMII or RGMI: Register 54 (SGMII/RGMI/SMII Control and Status Register) ■ General Purpose Input (GPI): Register 56 (General Purpose IO Register) <p>The interrupt signal from the DWC_gmac subsystem (sbd_intr_o) is high when this bit is high.</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
[25:23]	R	EB	<p>Error Bits</p> <p>This field indicates the type of error that caused a Bus Error, for example, error response on the AHB or AXI interface. This field is valid only when Bit 13 (FBI) is set. This field does not generate an interrupt.</p> <ul style="list-style-type: none"> ■ 0 0 0: Error during Rx DMA Write Data Transfer ■ 0 1 1: Error during Tx DMA Read Data Transfer ■ 1 0 0: Error during Rx DMA Descriptor Write Access ■ 1 0 1: Error during Tx DMA Descriptor Write Access ■ 1 1 0: Error during Rx DMA Descriptor Read Access ■ 1 1 1: Error during Tx DMA Descriptor Read Access <p>Note: 001 and 010 are reserved.</p>	3'b0
[22:20]	R	TS	<p>Transmit Process State</p> <p>This field indicates the Transmit DMA FSM state. This field does not generate an interrupt.</p> <ul style="list-style-type: none"> ■ 3'b000: Stopped; Reset or Stop Transmit Command issued ■ 3'b001: Running; Fetching Transmit Transfer Descriptor ■ 3'b010: Running; Waiting for status ■ 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO) ■ 3'b100: TIME_STAMP write state ■ 3'b101: Reserved for future use ■ 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow ■ 3'b111: Running; Closing Transmit Descriptor 	3'b0
[19:17]	R	RS	<p>Receive Process State</p> <p>This field indicates the Receive DMA FSM state. This field does not generate an interrupt.</p> <ul style="list-style-type: none"> ■ 3'b000: Stopped: Reset or Stop Receive Command issued ■ 3'b001: Running: Fetching Receive Transfer Descriptor ■ 3'b010: Reserved for future use ■ 3'b011: Running: Waiting for receive packet ■ 3'b100: Suspended: Receive Descriptor Unavailable ■ 3'b101: Running: Closing Receive Descriptor ■ 3'b110: TIME_STAMP write state ■ 3'b111: Running: Transferring the receive packet data from receive buffer to host memory 	3'b0
[16]	R/W	NIS	<p>Normal Interrupt Summary</p> <p>Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in Register 7 (Interrupt Enable Register):</p> <ul style="list-style-type: none"> ■ Register 5[0]: Transmit Interrupt ■ Register 5[2]: Transmit Buffer Unavailable ■ Register 5[6]: Receive Interrupt ■ Register 5[14]: Early Receive Interrupt <p>Only unmasked bits (interrupts for which interrupt enable is set in Register 7) affect the Normal Interrupt Summary bit.</p> <p>This is a sticky bit and must be cleared (by writing 1 to this bit) each time a corresponding bit, which causes NIS to be set, is cleared.</p>	1'b0
[15]	R/W	AIS	<p>Abnormal Interrupt Summary</p> <p>Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register 7 (Interrupt Enable Register):</p> <ul style="list-style-type: none"> ■ Register 5[1]: Transmit Process Stopped ■ Register 5[3]: Transmit Jabber Timeout 	1'b0

Bit	R/W	Symbol	Description	Reset Value
			<ul style="list-style-type: none"> ■ Register 5[4]: Receive FIFO Overflow ■ Register 5[5]: Transmit Underflow ■ Register 5[7]: Receive Buffer Unavailable ■ Register 5[8]: Receive Process Stopped ■ Register 5[9]: Receive Watchdog Timeout ■ Register 5[10]: Early Transmit Interrupt ■ Register 5[13]: Fatal Bus Error <p>Only unmasked bits affect the Abnormal Interrupt Summary bit.</p> <p>This is a sticky bit and must be cleared (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared.</p>	
[14]	R/W	ERI	<p>Early Receive Interrupt</p> <p>This bit indicates that the DMA filled the first data buffer of the packet. This bit is cleared when the software writes 1 to this bit or Bit 6 (RI) of this register is set (whichever occurs earlier).</p>	1'b0
[13]	R/W	FBI	<p>Fatal Bus Error Interrupt</p> <p>This bit indicates that a bus error occurred, as described in Bits [25:23]. When this bit is set, the corresponding DMA engine disables all of its bus accesses.</p>	1'b0
[12:11]	-	RESERVED	Reserved	2'b0
[10]	R/W	ETI	<p>Early Transmit Interrupt</p> <p>This bit indicates that the frame to be transmitted is fully transferred to the MTL Transmit FIFO.</p>	1'b0
[9]	R/W	RWT	<p>Receive Watchdog Timeout</p> <p>When set, this bit indicates that the Receive Watchdog Timer expired while receiving the current frame and the current frame is truncated after the watchdog timeout.</p>	1'b0
[8]	R/W	RPS	<p>Receive Process Stopped</p> <p>This bit is asserted when the Receive Process enters the Stopped state.</p>	1'b0
[7]	R/W	RU	<p>Receive Buffer Unavailable</p> <p>This bit indicates that the host owns the Next Descriptor in the Receive List and the DMA cannot acquire it. The Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, the Receive Process resumes when the next recognized incoming frame is received. This bit is set only when the previous Receive Descriptor is owned by the DMA.</p>	1'b0
[6]	R/W	RI	<p>Receive Interrupt</p> <p>This bit indicates that the frame reception is complete. When reception is complete, the Bit 31 of RDES1 (Disable Interrupt on Completion) is reset in the last Descriptor, and the specific frame status information is updated in the descriptor.</p> <p>The reception remains in the Running state.</p>	1'b0
[5]	R/W	UNF	<p>Transmit Underflow</p> <p>This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.</p>	1'b0
[4]	R/W	OVF	<p>Receive Overflow</p> <p>This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to the application, the overflow status is set in RDES0[11].</p>	1'b0
[3]	R/W	TJT	<p>Transmit Jabber Timeout</p> <p>This bit indicates that the Transmit Jabber Timer expired, which happens when the frame size exceeds 2,048 (10,240 bytes when the Jumbo frame is enabled). When the Jabber Timeout occurs, the transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
[2]	R/W	TU	<p>Transmit Buffer Unavailable</p> <p>This bit indicates that the host owns the Next Descriptor in the Transmit List and the DMA cannot acquire it. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions.</p> <p>To resume processing Transmit descriptors, the host should change the ownership of the descriptor by setting TDSEQ[31] and then issue a Transmit Poll Demand command.</p>	1'b0
Ethernet MAC DMA Register 6 (Operation Mode Register)				
Address : C006_1018h				
[31:27]	-	RESERVED	Reserved	5'b0
[26]	R/W	DT	<p>Disable Dropping of TCP/IP Checksum Error Frames</p> <p>When this bit is set, the MAC does not drop the frames which only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors only in the encapsulated payload. When this bit is reset, all error frames are dropped if the FEF bit is reset.</p> <p>If the IPC Full Checksum Offload Engine (Type 2) is disabled, this bit is reserved (RO with value 1'b0).</p>	1'b0
[25]	R/W	RSF	<p>Receive Store and Forward</p> <p>When this bit is set, the MTL reads a frame from the Rx FIFO only after the complete frame has been written to it, ignoring the RTC bits. When this bit is reset, the Rx FIFO operates in the cut-through mode, subject to the threshold specified by the RTC bits.</p>	1'b0
[24]	R/W	DFF	<p>Disable Flushing of Received Frames</p> <p>When this bit is set, the Rx DMA does not flush any frames because of the unavailability of receive descriptors or buffers as it does normally when this bit is reset.</p> <p>This bit is reserved (and RO) in the GMAC-MTL configuration.</p>	1'b0
[23]	R/W	RFA_2	<p>MSB of Threshold for Activating Flow Control</p> <p>If the DWC_gmac is configured for an Rx FIFO size of 8 KB or more, this bit (when set) provides additional threshold levels for activating the flow control in both half-duplex and full-duplex modes. This bit (as Most Significant Bit), along with the RFA (Bits [10:9]), gives the following thresholds for activating flow control:</p> <ul style="list-style-type: none"> ■ 100: Full minus 5 KB, that is, FULL — 5 KB ■ 101: Full minus 6 KB, that is, FULL — 6 KB ■ 110: Full minus 7 KB, that is, FULL — 7 KB ■ 111: Reserved <p>This bit is reserved (and RO) if the Rx FIFO is 4 KB or less deep.</p>	1'b0
[22]	R/W	RFD_2	<p>MSB of Threshold for Deactivating Flow Control</p> <p>If the DWC_gmac is configured for Rx FIFO size of 8 KB or more, this bit (when set) provides additional threshold levels for deactivating the flow control in both half-duplex and full-duplex modes. This bit (as Most Significant Bit) along with the RFD (Bits [12:11]) gives the following thresholds for deactivating flow control:</p> <ul style="list-style-type: none"> ■ 100: Full minus 5 KB, that is, FULL — 5 KB ■ 101: Full minus 6 KB, that is, FULL — 6 KB ■ 110: Full minus 7 KB, that is, FULL — 7 KB ■ 111: Reserved <p>This bit is reserved (and RO) if the Rx FIFO is 4 KB or less deep.</p>	1'b0
[21]	R/W	TSF	<p>Transmit Store and Forward</p> <p>When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Bits [16:14] are ignored. This bit should be changed only when the transmission is stopped.</p>	1'b0
[20]	R/W	FTF	Flush Transmit FIFO	1'b0

Bit	R/W	Symbol	Description	Reset Value								
			<p>When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost or flushed. This bit is cleared internally when the flushing operation is complete. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt frame transmission.</p> <p>Note:</p> <p>The flush operation is complete only when the Tx FIFO is emptied of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active.</p>									
[19:17]	-	RESERVED	Reserved	3'b0								
[16:14]	R/W	TTC	<p>Transmit Threshold Control</p> <p>These bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when Bit 21 (TSF) is reset.</p> <table> <tr> <td>■ 000: 64</td> <td>■ 001: 128</td> </tr> <tr> <td>■ 010: 192</td> <td>■ 011: 256</td> </tr> <tr> <td>■ 100: 40</td> <td>■ 101: 32</td> </tr> <tr> <td>■ 110: 24</td> <td>■ 111: 16</td> </tr> </table>	■ 000: 64	■ 001: 128	■ 010: 192	■ 011: 256	■ 100: 40	■ 101: 32	■ 110: 24	■ 111: 16	3'b0
■ 000: 64	■ 001: 128											
■ 010: 192	■ 011: 256											
■ 100: 40	■ 101: 32											
■ 110: 24	■ 111: 16											
[13]	R/W	ST	<p>Start or Stop Transmission Command</p> <p>When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register 4 (Transmit Descriptor List Address Register), or from the position retained when transmission was stopped previously. If the DMA does not own the current descriptor, transmission enters the Suspended state and Bit 2 (Transmit Buffer Unavailable) of Register 5 (Status Register) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting Register 4 (Transmit Descriptor List Address Register), then the DMA behavior is unpredictable.</p> <p>When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and it becomes the current position when transmission is restarted. To change the list address, you need to program Register 4 (Transmit Descriptor List Address Register) with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current frame is complete or the transmission is in the Suspended state.</p>	1'b0								
[12:11]	R/W	RFD	<p>Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)</p> <p>These bits control the threshold (FillLevel of Rx FIFO) at which the flow control is de-asserted after activation.</p> <table> <tr> <td>■ 00: Full minus 1 KB, that is, FULL — 1 KB</td> </tr> <tr> <td>■ 01: Full minus 2 KB, that is, FULL — 2 KB</td> </tr> <tr> <td>■ 10: Full minus 3 KB, that is, FULL — 3 KB</td> </tr> <tr> <td>■ 11: Full minus 4 KB, that is, FULL — 4 KB</td> </tr> </table> <p>The de-assertion is effective only after flow control is asserted. If the Rx FIFO is 8 KB or more, an additional Bit (RFD_2) is used for more threshold levels as described in Bit 22. These bits are reserved and read-only when the Rx FIFO depth is less than 4 KB.</p> <p>Note: For proper flow control, the value programmed in the "RFD_2, RFD" fields should be equal to or more than the value programmed in the "RFA_2, RFA" fields.</p>	■ 00: Full minus 1 KB, that is, FULL — 1 KB	■ 01: Full minus 2 KB, that is, FULL — 2 KB	■ 10: Full minus 3 KB, that is, FULL — 3 KB	■ 11: Full minus 4 KB, that is, FULL — 4 KB	2'b0				
■ 00: Full minus 1 KB, that is, FULL — 1 KB												
■ 01: Full minus 2 KB, that is, FULL — 2 KB												
■ 10: Full minus 3 KB, that is, FULL — 3 KB												
■ 11: Full minus 4 KB, that is, FULL — 4 KB												
[10:9]	R/W	RFA	<p>Threshold for Activating Flow Control (in half-duplex and full-duplex modes)</p> <p>These bits control the threshold (Fill level of Rx FIFO) at which the flow control is activated.</p> <table> <tr> <td>■ 00: Full minus 1 KB, that is, FULL—1KB.</td> </tr> </table>	■ 00: Full minus 1 KB, that is, FULL—1KB.	2'b0							
■ 00: Full minus 1 KB, that is, FULL—1KB.												

Bit	R/W	Symbol	Description	Reset Value
			<ul style="list-style-type: none"> ■ 01: Full minus 2 KB, that is, FULL—2KB. ■ 10: Full minus 3 KB, that is, FULL—3KB. ■ 11: Full minus 4 KB, that is, FULL—4KB. <p>These values are applicable only to Rx FIFOs of 4 KB or more and when Bit 8 (EFC) is set high. If the Rx FIFO is 8 KB or more, an additional Bit (RFA_2) is used for more threshold levels as described in Bit 23. These bits are reserved and read-only when the depth of Rx FIFO is less than 4 KB.</p> <p>Note: When FIFO size is exactly 4 KB, although the DWC_gmac allows you to program the value of these bits to 11, the software should not program these bits to 2b11. The value 2b11 means flow control on FIFO empty condition.</p>	
[8]	R/W	EFC	<p>Enable HW Flow Control</p> <p>When this bit is set, the flow control signal operation based on the fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled. This bit is not used (reserved and always reset) when the Rx FIFO is less than 4 KB.</p>	1'b0
[7]	R/W	FEF	<p>Forward Error Frames</p> <p>When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, or overflow). However, if the start byte (write) pointer of a frame is already transferred to the read controller side (in Threshold mode), then the frame is not dropped.</p> <p>In the GMAC-MTL configuration in which the Frame Length FIFO is also enabled during core configuration, the Rx FIFO drops the error frames if that frame's start byte is not transferred (output) on the ARI bus.</p> <p>When the FEF bit is set, all frames except runt error frames are forwarded to the DMA. If the Bit 25 (RSF) is set and the Rx FIFO overflows when a partial frame is written, then the frame is dropped irrespective of the FEF bit setting. However, if the Bit 25 (RSF) is reset and the Rx FIFO overflows when a partial frame is written, then a partial frame may be forwarded to the DMA.</p> <p>Note: When FEF bit is reset, the giant frames are dropped if the giant frame status is given in Rx Status in the following configurations:</p> <ul style="list-style-type: none"> ■ The IP checksum engine (Type 1) and full checksum offload engine (Type 2) are not selected. ■ The advanced timestamp feature is not selected but the extended status is selected. The extended status is available with the following features: <ul style="list-style-type: none"> - L3-L4 filter in GMAC-CORE or GMAC-MTL configurations - Full checksum offload engine (Type 2) with enhanced descriptor format in the GMAC-DMA, GMAC-AHB, or GMAC-AXI configurations. 	1'b0
[6]	R/W	FUF	<p>Forward Undersized Good Frames</p> <p>When set, the Rx FIFO forwards Undersized frames (that is, frames with no Error and length less than 64 bytes) including pad-bytes and CRC.</p> <p>When reset, the Rx FIFO drops all frames of less than 64 bytes, unless a frame is already transferred because of the lower value of Receive Threshold, for example, RTC = 01.</p>	1'b0
[5]	R/W	DGF	<p>Drop Giant Frames</p> <p>When set, the MAC drops the received giant frames in the Rx FIFO, that is, frames that are larger than the computed giant frame limit. When reset, the MAC does not drop the giant frames in the Rx FIFO.</p> <p>Note: This bit is available in the following configurations in which the giant frame status is not provided in Rx status and giant frames are not dropped by default:</p> <ul style="list-style-type: none"> ■ Configurations in which IP Checksum Offload (Type 1) is selected in Rx ■ Configurations in which the IPC Full Checksum Offload Engine (Type 2) is selected in Rx with normal descriptor format ■ Configurations in which the Advanced Timestamp feature is selected <p>In all other configurations, this bit is not used (reserved and always reset).</p>	1'b0
[4:3]	R/W	RTC	Threshold Control	2'b0

Bit	R/W	Symbol	Description	Reset Value
			<p>These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with length less than the threshold are automatically transferred.</p> <p>The value of 11 is not applicable if the configured Receive FIFO size is 128 bytes.</p> <p>These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1.</p> <ul style="list-style-type: none"> ■ 00: 64 ■ 01: 32 ■ 10: 96 ■ 11: 128 	
[2]	R/W	OSF	<p>Operate on Second Frame</p> <p>When this bit is set, it instructs the DMA to process the second frame of the Transmit data even before the status for the first frame is obtained.</p>	2'b0
[1]	R/W	SR	<p>Start or Stop Receive</p> <p>When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes the incoming frames. The descriptor acquisition is attempted from the current position in the list, which is the address set by the Register 3 (Receive Descriptor List Address Register) or the position retained when the Receive process was previously stopped. If the DMA does not own the descriptor, reception is suspended and Bit 7 (Receive Buffer Unavailable) of Register 5 (Status Register) is set. The Start Receive command is effective only when the reception has stopped. If the command is issued before setting Register 3 (Receive Descriptor List Address Register), the DMA behavior is unpredictable.</p> <p>When this bit is cleared, the Rx DMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.</p>	2'b0
[0]	-	RESERVED	Reserved	1'b0

Ethernet MAC DMA Register 7 (Interrupt Enable Register)

Address : C006_101Ch

[31:17]	-	RESERVED	Reserved	15'b0
[16]	R/W	NIE	<p>Normal Interrupt Summary Enable</p> <p>When this bit is set, normal interrupt summary is enabled. When this bit is reset, normal interrupt summary is disabled. This bit enables the following interrupts in Register 5 (Status Register):</p> <ul style="list-style-type: none"> ■ Register 5[0]: Transmit Interrupt ■ Register 5[2]: Transmit Buffer Unavailable ■ Register 5[6]: Receive Interrupt ■ Register 5[14]: Early Receive Interrupt 	1'b0
[15]	R/W	AIE	<p>Abnormal Interrupt Summary Enable</p> <p>When this bit is set, abnormal interrupt summary is enabled. When this bit is reset, the abnormal interrupt summary is disabled. This bit enables the following interrupts in Register 5 (Status Register):</p> <ul style="list-style-type: none"> ■ Register 5[1]: Transmit Process Stopped ■ Register 5[3]: Transmit Jabber Timeout ■ Register 5[4]: Receive Overflow ■ Register 5[5]: Transmit Underflow ■ Register 5[7]: Receive Buffer Unavailable ■ Register 5[8]: Receive Process Stopped ■ Register 5[9]: Receive Watchdog Timeout ■ Register 5[10]: Early Transmit Interrupt 	1'b0

Bit	R/W	Symbol	Description	Reset Value
			■ Register 5[13]: Fatal Bus Error	
[14]	R/W	ERE	Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Early Receive Interrupt is enabled. When this bit is reset, the Early Receive Interrupt is disabled.	1'b0
[13]	R/W	FBE	Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, the Fatal Bus Error Enable Interrupt is disabled.	1'b0
[12:11]	-	RESERVED	Reserved	2'b0
[10]	R/W	ETE	Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (Bit 15), the Early Transmit Interrupt is enabled. When this bit is reset, the Early Transmit Interrupt is disabled.	1'b0
[9]	R/W	RWE	Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout Interrupt is disabled.	1'b0
[8]	R/W	RSE	Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped Interrupt is disabled.	1'b0
[7]	R/W	RUE	Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.	1'b0
[6]	R/W	RIE	Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled.	1'b0
[5]	R/W	UNE	Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Underflow Interrupt is enabled. When this bit is reset, the Underflow Interrupt is disabled.	1'b0
[4]	R/W	OVE	Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Receive Overflow Interrupt is enabled. When this bit is reset, the Overflow Interrupt is disabled.	1'b0
[3]	R/W	TJE	Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, the Transmit Jabber Timeout Interrupt is disabled.	1'b0
[2]	R/W	TUE	Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable Interrupt is disabled.	1'b0
[1]	R/W	TSE	Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (Bit 15), the Transmission Stopped Interrupt is enabled. When this bit is reset, the Transmission Stopped Interrupt is disabled.	1'b0
[0]	R/W	TIE	Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (Bit 16), the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled.	1'b0
Ethernet MAC DMA Register 8 (Missed Frame and Buffer Overflow Control Register)				

Bit	R/W	Symbol	Description	Reset Value
Address : C006_1020h				
[31:29]	-	RESERVED	Reserved	3'b0
[28]	R/W	OVFCNTOVF	Overflow Bit for FIFO Overflow Counter This bit is set every time the Overflow Frame Counter (Bits[27:17]) overflows, that is, the Rx FIFO overflows with the overflow frame counter at maximum value. In such a scenario, the overflow frame counter is reset to all-zeros and this bit indicates that the rollover happened.	1'b0
[27:17]	R/W	OVFFRMCNT	Overflow Frame Counter This field indicates the number of frames missed by the application. This counter is incremented each time the MTL FIFO overflows. The counter is cleared when this register is read with mci_be_i[2] at 1'b1.	11'b0
[16]	R/W	MISCNTOVF	Overflow Bit for Missed Frame Counter This bit is set every time Missed Frame Counter (Bits[15:0]) overflows, that is, the DMA discards an incoming frame because of the Host Receive Buffer being unavailable with the missed frame counter at maximum value. In such a scenario, the Missed frame counter is reset to all-zeros and this bit indicates that the rollover happened.	1'b0
[15:0]	R/W	MISFRMCNT	Missed Frame Counter This field indicates the number of frames missed by the controller because of the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.	16'b0
RESERVED				
Address : C006_1024h ~ C006_1044h				
Ethernet MAC DMA Register 18 (Current Host Transmit Descriptor Register)				
Address : C006_1048h				
[31:0]	R/W	CURTDESAPTR	Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.	32'b0
Ethernet MAC DMA Register 19 (Current Host Receive Descriptor Register)				
Address : C006_104Ch				
[31:0]	R/W	CURRDESAPTR	Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.	32'b0
Ethernet MAC DMA Register 20 (Current Host Transmit Buffer Address Register)				
Address : C006_1050h				
[31:0]	R/W	CURTBUFAPTR	Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by the DMA during operation.	32'b0
Ethernet MAC DMA Register 21 (Current Host Receive Buffer Address Register)				
Address : C006_1054h				
[31:0]	R/W	CURRBUFAPTR	Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by the DMA during operation. operation.	32'b0

16.2.2 MAC Core Register Description

Bit	R/W	Symbol	Description	Reset Value
Ethernet MAC Register 0 (MAC Configuration Register)				
Address : C006_0000h				
[31]	-	RESERVED	Reserved	1'b0
[30:28]	R/W	SARC	<p>Source Address Insertion or Replacement Control</p> <p>This field controls the source address insertion or replacement for all transmitted frames. Bit 30 specifies which MAC Address register (0 or 1) is used for source address insertion or replacement based on the values of Bits [29:28]:</p> <ul style="list-style-type: none"> ■ 2b0x: The input signals mti_sa_ctrl_i and ati_sa_ctrl_i control the SA field generation. ■ 2b10: <ul style="list-style-type: none"> - If Bit 30 is set to 0, the MAC inserts the content of the MAC Address 0 registers (registers 16 and 17) in the SA field of all transmitted frames. - If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected during core configuration, the MAC inserts the content of the MAC Address 1 registers (registers 18 and 19) in the SA field of all transmitted frames. ■ 2b11: <ul style="list-style-type: none"> - If Bit 30 is set to 0, the MAC replaces the content of the MAC Address 0 registers (registers 16 and 17) in the SA field of all transmitted frames. - If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected during core configuration, the MAC replaces the content of the MAC Address 1 registers (registers 18 and 19) in the SA field of all transmitted frames. <p>Note:</p> <ul style="list-style-type: none"> ■ Changes to this field take effect only on the start of a frame. If you write this register field when a frame is being transmitted, only the subsequent frame can use the updated value, that is, the current frame does not use the updated value. ■ These bits are reserved and RO when the Enable SA, VLAN, and CRC Insertion on TX feature is not selected during core configuration. 	3'b0
[27]	R/W	TWOKPE	<p>IEEE 802.3as Support for 2K Packets</p> <p>When set, the MAC considers all frames, with up to 2,000 bytes length, as normal packets.</p> <p>When Bit 20 (JE) is not set, the MAC considers all received frames of size more than 2K bytes as Giant frames. When this bit is reset and Bit 20 (JE) is not set, the MAC considers all received frames of size more than 1,518 bytes (1,522 bytes for tagged) as Giant frames. When Bit 20 is set, setting this bit has no effect on Giant Frame status.</p>	1'b0
[26]	R/W	SFTERR	<p>SMII Force Transmit Error</p> <p>When set, this bit indicates to the PHY to force a transmit error in the SMII frame being transmitted. This bit is reserved if the SMII PHY port is not selected during core configuration.</p>	1'b0
[25]	R/W	CST	<p>CRC Stripping for Type Frames</p> <p>When this bit is set, the last 4 bytes (FCS) of all frames of Ether type (Length/Type field greater than or equal to 1,536) are stripped and dropped before forwarding the frame to the application. This function is not valid when the IP Checksum Engine (Type 1) is enabled in the MAC receiver. This function is valid when Type 2 Checksum Offload Engine is enabled.</p>	1'b0
[24]	R/W	TC	<p>Transmit Configuration in RGMII, SGMII, or SMII</p> <p>When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. When this bit is reset, no such information is driven to the PHY. This bit is reserved (and RO) if the RGMII, SMII, or SGMII PHY port is not selected during core configuration.</p>	1'b0
[23]	R/W	WD	<p>Watchdog Disable</p> <p>When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive frames of up to 16,384 bytes.</p> <p>When this bit is reset, the MAC does not allow a receive frame which more than 2,048 bytes</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
			(10,240 if JE is set high) or the value programmed in Register 55 (Watchdog Timeout Register). The MAC cuts off any bytes received after the watchdog limit number of bytes.	
[22]	R/W	JD	Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer frames of up to 16,384 bytes. When this bit is reset, the MAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.	1'b0
[21]	R/W	BE	Frame Burst Enable When this bit is set, the MAC allows frame bursting during transmission in the GMII half-duplex mode. This bit is reserved (and RO) in the 10/100 Mbps only or full-duplex-only configurations.	1'b0
[20]	R/W	JE	Jumbo Frame Enable When this bit is set, the MAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status.	1'b0
[19:17]	R/W	IFG	Inter-Frame Gap These bits control the minimum IFG between frames during transmission. <ul style="list-style-type: none">■ 000: 96 bit times■ 001: 88 bit times■ 010: 80 bit times■ ...■ 111: 40 bit times In the half-duplex mode, the minimum IFG can be configured only for 64 bit times (IFG = 100). Lower values are not considered. In the 1000-Mbps mode, the minimum IFG supported is 64 bit times (and above) in the GMAC-CORE configuration and 80 bit times (and above) in other configurations.	3'b0
[16]	R/W	DCRS	Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in the half-duplex mode. This request results in no errors generated because of Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors because of Carrier Sense and can even abort the transmissions. This bit is reserved (and RO) in the full-duplex-only configurations.	1'b0
[15]	R/W	PS	Port Select This bit selects the Ethernet line speed. <ul style="list-style-type: none">■ 0: For 1000 Mbps operations■ 1: For 10 or 100 Mbps operations In 10 or 100 Mbps operations, this bit, along with FES bit, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only with the appropriate value. In default 10/100/1000 Mbps configuration, this bit is R_W. The mac_portselect_o or mac_speed_o[1] signal reflects the value of this bit.	1'b0
[14]	R/W	FES	Speed This bit selects the speed in the MII, RMII, SMII, RGMII, SGMII, or RevMII interface: <ul style="list-style-type: none">■ 0: 10 Mbps■ 1: 100 Mbps This bit is reserved (RO) by default and is enabled only when the parameter SPEED_SELECT = Enabled. This bit generates link speed encoding when Bit 24 (TC) is set in the RGMII, SMII, or SGMII mode. This bit is always enabled for RGMII, SGMII, SMII, or RevMII interface. In configurations with RGMII, SGMII, SMII, or RevMII interface, this bit is driven as an output signal (mac_speed_o[0]) to reflect the value of this bit in the mac_speed_o signal. In configurations with RMII, MII, or GMII interface, you can optionally drive this bit as an output signal (mac_speed_o[0]) to reflect its value in the mac_speed_o signal.	1'b0
[13]	R/W	DO	Disable Receive Own When this bit is set, the MAC disables the reception of frames when the phy_txen_o is asserted	1'b0

Bit	R/W	Symbol	Description	Reset Value				
			<p>in the half-duplex mode.</p> <p>When this bit is reset, the MAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the MAC is operating in the full-duplex mode. This bit is reserved (RO with default value) if the MAC is configured for the full-duplex-only operation.</p>					
[12]	R/W	LM	<p>Loopback Mode</p> <p>When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Receive clock input (<code>clk_rx_i</code>) is required for the loopback to work properly, because the Transmit clock is not looped-back internally.</p>	1'b0				
[11]	R/W	DM	<p>Duplex Mode</p> <p>When this bit is set, the MAC operates in the full-duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in the full-duplex-only configuration.</p>	1'b0				
[10]	R/W	IPC	<p>Checksum Offload</p> <p>When this bit is set, the MAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25–26 or 29–30 (VLANtagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The MAC also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected).</p> <p>When this bit is reset, this function is disabled.</p> <p>When Type 2 COE is selected, this bit, when set, enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared.</p> <p>If the IP Checksum Offload feature is not enabled during core configuration, this bit is reserved (RO with default value).</p>	1'b0				
[9]	R/W	DR	<p>Disable Retry</p> <p>When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current frame transmission and reports a Frame Abort with excessive collision error in the transmit frame status.</p> <p>When this bit is reset, the MAC attempts retries based on the settings of the BL field (Bits [6:5]). This bit is applicable only in the half-duplex mode and is reserved (RO with default value) in the full-duplex-only configuration.</p>	1'b0				
[8]	R/W	LUD	<p>Link Up or Down</p> <p>This bit indicates whether the link is up or down during the transmission of configuration in the RGMII, SGMII, or SMII interface:</p> <table style="margin-left: 20px;"> <tr> <td>■ 0: Link Down</td> <td>■ 1: Link Up</td> </tr> </table> <p>This bit is reserved (RO with default value) and is enabled when the RGMII, SGMII, or SMII interface is enabled during core configuration.</p>	■ 0: Link Down	■ 1: Link Up	1'b0		
■ 0: Link Down	■ 1: Link Up							
[7]	R/W	ACS	<p>Automatic Pad or CRC Stripping</p> <p>When this bit is set, the MAC strips the Pad or FCS field on the incoming frames only if the value of the length field is less than 1,536 bytes. All received frames with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field.</p> <p>When this bit is reset, the MAC passes all incoming frames, without modifying them, to the Host.</p> <p>Note: For information about how the settings of Bit 23 (CST) and this bit impact the frame length</p>	1'b0				
[6:5]	R/W	BL	<p>Back-Off Limit</p> <p>The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only in the half-duplex mode and is reserved (RO) in the full-duplex-only configuration.</p> <table style="margin-left: 20px;"> <tr> <td>■ 00: $k = \min(n, 10)$</td> <td>■ 01: $k = \min(n, 8)$</td> </tr> <tr> <td>■ 10: $k = \min(n, 4)$</td> <td>■ 11: $k = \min(n, 1)$</td> </tr> </table>	■ 00: $k = \min(n, 10)$	■ 01: $k = \min(n, 8)$	■ 10: $k = \min(n, 4)$	■ 11: $k = \min(n, 1)$	2'b0
■ 00: $k = \min(n, 10)$	■ 01: $k = \min(n, 8)$							
■ 10: $k = \min(n, 4)$	■ 11: $k = \min(n, 1)$							

Bit	R/W	Symbol	Description	Reset Value
			where n = retransmission attempt. The random integer r takes the value in the range $0 \leq r < 2k$	
[4]	R/W	DC	<p>Deferral Check</p> <p>When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status, when the transmit state machine is deferred for more than 24,288 bit times in the 10 or 100 Mbps mode.</p> <p>If the MAC is configured for 1000 Mbps operation or if the Jumbo frame mode is enabled in the 10 or 100 Mbps mode, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII.</p> <p>The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and then the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0 and it is restarted.</p> <p>When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode and is reserved (RO) in the full-duplex-only configuration.</p>	1'b0
[3]	R/W	TE	<p>Transmitter Enable</p> <p>When this bit is set, the transmit state machine of the MAC is enabled for transmission on the GMII or MII. When this bit is reset, the MAC transmit state machine is disabled after the completion of the transmission of the current frame, and does not transmit any further frames.</p>	1'b0
[2]	R/W	RE	<p>Receiver Enable</p> <p>When this bit is set, the receiver state machine of the MAC is enabled for receiving frames from the GMII or MII. When this bit is reset, the MAC receive state machine is disabled after the completion of the reception of the current frame, and does not receive any further frames from the GMII or MII.</p>	1'b0
[1:0]	R/W	PRELEN	<p>Preamble Length for Transmit frames</p> <p>These bits control the number of preamble bytes that are added to the beginning of every Transmit frame. The preamble reduction occurs only when the MAC is operating in the full-duplex mode.</p> <ul style="list-style-type: none"> ■ 2b00: 7 bytes of preamble ■ 2b01: 5 bytes of preamble ■ 2b10: 3 bytes of preamble ■ 2b11: Reserved 	2'b0
Ethernet MAC Register 1 (MAC Frame Filter Register)				
Address : C006_0004h				
[31]	R/W	RA	<p>Receive All</p> <p>When this bit is set, the MAC Receiver module passes all received frames, irrespective of whether they pass the address filter or not, to the Application. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word.</p> <p>When this bit is reset, the Receiver module passes only those frames to the Application that pass the SA or DA address filter.</p>	1'b0
[30:22]	-	RESERVED	Reserved	9'b0
[21]	R/W	DNTU	<p>Drop non-TCP/UDP over IP Frames</p> <p>When set, this bit enables the MAC to drop the non-TCP or UDP over IP frames. The MAC forward only those frames that are processed by the Layer 4 filter. When reset, this bit enables the MAC to forward all non-TCP or UDP over IP frames.</p> <p>If the Layer 3 and Layer 4 Filtering feature is not selected during core configuration, this bit is reserved (RO with default value).</p>	1'b0
[20]	R/W	IPFE	<p>Layer 3 and Layer 4 Filter Enable</p> <p>When set, this bit enables the MAC to drop frames that do not match the enabled Layer 3 and Layer 4 filters. If Layer 3 or Layer 4 filters are not enabled for matching, this bit does not have any effect.</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
			When reset, the MAC forwards all frames irrespective of the match status of the Layer 3 and Layer 4 fields. If the Layer 3 and Layer 4 Filtering feature is not selected during core configuration, this bit is reserved (RO with default value).	
[19:17]	-	RESERVED	Reserved	3'b0
[16]	R/W	VTFE	VLAN Tag Filter Enable When set, this bit enables the MAC to drop VLAN tagged frames that do not match the VLAN Tag comparison. When reset, the MAC forwards all frames irrespective of the match status of the VLAN Tag.	1'b0
[15:11]	-	RESERVED	Reserved	5'b0
[10]	R/W	HPF	Hash or Perfect Filter When this bit is set, it configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by the HMC or HUC bits. When this bit is low and the HUC or HMC bit is set, the frame is passed only if it matches the Hash filter. This bit is reserved (and RO) if the Hash filter is not selected during core configuration.	1'b0
[9]	R/W	SAF	Source Address Filter Enable When this bit is set, the MAC compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison fails, the MAC drops the frame. When this bit is reset, the MAC forwards the received frame to the application with updated SAF bit of the Rx Status depending on the SA address comparison. Note: According to the IEEE specification, Bit 47 of the SA is reserved and set to 0. However, in DWC_gmac, the MAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA..	1'b0
[8]	R/W	SAIF	SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers are marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers are marked as failing the SA Address filter.	1'b0
[7:6]	R/W	PCF	Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast Pause frames). ■ 00: MAC filters all control frames from reaching the application. ■ 01: MAC forwards all control frames except Pause frames to application even if they fail the Address filter. ■ 10: MAC forwards all control frames to application even if they fail the Address Filter. ■ 11: MAC forwards control frames that pass the Address Filter. The following conditions should be true for the Pause frames processing: ■ Condition 1: The MAC is in the full-duplex mode and flow control is enabled by setting Bit 2 (RFE) of Register 6 (Flow Control Register) to 1. ■ Condition 2: The destination address (DA) of the received frame matches the special multicast address or the MAC Address 0 when Bit 3 (UP) of the Register 6 (Flow Control Register) is set. ■ Condition 3: The Type field of the received frame is 0x8808 and the OPCODE field is 0x0001. Note: This field should be set to 01 only when the Condition 1 is true, that is, the MAC is programmed to operate in the full-duplex mode and the RFE bit is enabled. Otherwise, the Pause frame filtering may be inconsistent. When Condition 1 is false, the Pause frames are considered as generic control frames. Therefore, to pass all control frames (including Pause frames) when the full-duplex mode and flow control is not enabled, you should set the PCF field to 10 or 11 (as required by the application).	2'b0
[5]	R/W	DBF	Disable Broadcast Frames	1'b0

Bit	R/W	Symbol	Description	Reset Value
			When this bit is set, the AFM module blocks all incoming broadcast frames. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast frames.	
[4]	R/W	PM	Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.	1'b0
[3]	R/W	DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.	1'b0
[2]	R/W	HMC	Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers. If Hash Filter is not selected during core configuration, this bit is reserved (and RO).	1'b0
[1]	R/W	HUC	Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers. If Hash Filter is not selected during core configuration, this bit is reserved (and RO).	1'b0
[0]	R/W	PR	Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames irrespective of the destination or source address. The SA or DA Filter Fails status bits of the Receive Status Word are always cleared when PR is set.	1'b0

Ethernet MAC Register 2 (Hash Table High Register)

Address : C006_0008h

[31:0]	R/W	HTH	Hash Table High This field contains the upper 32 bits of the Hash table.	32'b0
--------	-----	------------	--	-------

Ethernet MAC Register 3 (Hash Table Low Register)

Address : C006_000Ch

[31:0]	R/W	HTL	Hash Table Low This field contains the lower 32 bits of the Hash table.	32'b0
--------	-----	------------	---	-------

Ethernet MAC Register 4 (GMII AddressRegister)

Address : C006_0010h

[31:16]	-	RESERVED	Reserved	16'b0
[15:11]	R/W	PA	Physical Layer Address This field indicates which of the 32 possible PHY devices are being accessed. For RevMII, this field gives the PHY Address of the RevMII module.	5'b0
[10:6]	R/W	GR	GMII Register These bits select the desired GMII register in the selected PHY device. For RevMII, these bits select the desired CSR register in the RevMII Registers set.	5'b0
[5:2]	R/W	CR	CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design. The CSR clock corresponding to different GMAC configurations. The suggested range of CSR clock frequency applicable for each value (when Bit[5] = 0)	4'b0

Bit	R/W	Symbol	Description	Reset Value
			<p>ensures that the MDC clock is approximately between the frequency range 1.0 MHz–2.5 MHz.</p> <ul style="list-style-type: none"> ■ 0000: The CSR clock frequency is 60–100 MHz and the MDC clock frequency is CSR clock/42. ■ 0001: The CSR clock frequency is 100–150 MHz and the MDC clock frequency is CSR clock/62. ■ 0010: The CSR clock frequency is 20–35 MHz and the MDC clock frequency is CSR clock/16. ■ 0011: The CSR clock frequency is 35–60 MHz and the MDC clock frequency is CSR clock/26. ■ 0100: The CSR clock frequency is 150–250 MHz and the MDC clock frequency is CSR clock/102. ■ 0101: The CSR clock frequency is 250–300 MHz and the MDC clock frequency is CSR clock/124. ■ 0110, 0111: Reserved <p>When Bit 5 is set, you can achieve higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE Std 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, then the resultant MDC clock is of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Program the following values only if the interfacing chips support faster MDC clocks.</p> <ul style="list-style-type: none"> ■ 1000: CSR clock/4 ■ 1001: CSR clock/6 ■ 1010: CSR clock/8 ■ 1011: CSR clock/10 ■ 1100: CSR clock/12 ■ 1101: CSR clock/14 ■ 1110: CSR clock/16 ■ 1111: CSR clock/18 <p>These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.</p>	
[1]	R/W	GW	<p>GMII Write</p> <p>When set, this bit indicates to the PHY or RevMII that this is a Write operation using the GMII Data register. If this bit is not set, it indicates that this is a Read operation, that is, placing the data in the GMII Data register.</p>	1'b0
[0]	R/W	GB	<p>GMII Busy</p> <p>This bit should read logic 0 before writing to Register 4 and Register 5. During a PHY or RevMII register access, the software sets this bit to 1'b1 to indicate that a Read or Write access is in progress.</p> <p>Register 5 is invalid until this bit is cleared by the MAC. Therefore, Register 5 (GMII Data) should be kept valid until the MAC clears this bit during a PHY Write operation. Similarly for a read operation, the contents of Register 5 are not valid until this bit is cleared.</p> <p>The subsequent read or write operation should happen only after the previous operation is complete. Because there is no acknowledgment from the PHY to MAC after a read or write operation is completed, there is no change in the functionality of this bit even when the PHY is not present.</p>	1'b0
Ethernet MAC Register 5 (GMII Data Register)				
Address : C006_0014h				
[31:16]	-	RESERVED	Reserved	16'b0
[15:0]	R/W	GD	<p>GMII Data</p> <p>This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.</p>	16'b0
Ethernet MAC Register 6 (Flow Control Register)				

Bit	R/W	Symbol	Description	Reset Value
Address : C006_0018h				
[31:16]	R/W	PT	<p>Pause Time</p> <p>This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain</p>	16'b0
[15:8]	-	RESERVED	Reserved	8'b0
[7]	R/W	DZQP	<p>Disable Zero-Quanta Pause</p> <p>When this bit is set, it disables the automatic generation of the Zero-Quanta Pause frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal <code>sbd_flowctrl_i</code>/<code>mti_flowctrl_i</code>).</p> <p>When this bit is reset, normal operation with automatic Zero-Quanta Pause frame generation is enabled.</p>	1'b0
[6]	-	RESERVED	Reserved	1'b0
[5:4]	R/W	PLT	<p>Pause Low Threshold</p> <p>This field configures the threshold of the Pause timer at which the input flow control signal <code>mti_flowctrl_i</code> (or <code>sbd_flowctrl_i</code>) is checked for automatic retransmission of the Pause frame.</p> <p>The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second Pause frame is automatically transmitted if the <code>mti_flowctrl_i</code> signal is asserted at 228 (256 – 28) slot times after the first Pause frame is transmitted.</p> <p>The following list provides the threshold values for different values:</p> <ul style="list-style-type: none"> ■ 00: The threshold is Pause time minus 4 slot times (PT – 4 slot times). ■ 01: The threshold is Pause time minus 28 slot times (PT – 28 slot times). ■ 10: The threshold is Pause time minus 144 slot times (PT – 144 slot times). ■ 11: The threshold is Pause time minus 256 slot times (PT – 256 slot times). <p>The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.</p>	2b0
[3]	R/W	UP	<p>Unicast Pause Frame Detect</p> <p>A pause frame is processed when it has the unique multicast address specified in the IEEE Std 802.3. When this bit is set, the MAC can also detect Pause frames with unicast address of the station. This unicast address should be as specified in the MAC Address0 High Register and MAC Address0 Low Register.</p> <p>When this bit is reset, the MAC only detects Pause frames with unique multicast address.</p> <p>Note: The MAC does not process a Pause frame if the multicast address of received frame is different from the unique multicast address.</p>	1'b0
[2]	R/W	PFE	<p>Receive Flow Control Enable</p> <p>When this bit is set, the MAC decodes the received Pause frame and disables its transmitter for a specified (Pause) time. When this bit is reset, the decode function of the Pause frame is disabled.</p>	1'b0
[1]	R/W	TFE	<p>Transmit Flow Control Enable</p> <p>In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause frames.</p> <p>In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled.</p>	1'b0
[0]	R/W	FCP_BPA	<p>Flow Control Busy or Backpressure Activate</p> <p>This bit initiates a Pause frame in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set.</p> <p>In the full-duplex mode, this bit should be read as 1'b0 before writing to the Flow Control register.</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
			To initiate a Pause frame, the Application must set this bit to 1'b1. During a transfer of the Control Frame, this bit continues to be set to signify that a frame transmission is in progress. After the completion of Pause frame transmission, the MAC resets this bit to 1'b0. The Flow Control register should not be written to until this bit is cleared. In the half-duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the MAC. During backpressure, when the MAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled.	
Ethernet MAC Register 7 (VALN Tag Register)				
Address : C006_001Ch				
[31:20]	-	RESERVED	Reserved	12'b0
[19]	R/W	VTHM	VLAN Tag Hash Table Match Enable When set, the most significant four bits of the VLAN tag's CRC are used to index the content of Register 354 (VLAN Hash Table Register). A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the frame matched the VLAN hash table. When Bit 16 (ETV) is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison whereas when ETV is reset, the CRC of the 16-bit VLAN tag is used for comparison. When reset, the VLAN Hash Match operation is not performed. If the VLAN Hash feature is not enabled during core configuration, this bit is reserved (RO with default value).	1'b0
[18]	R/W	ESVL	Enable S-VLAN When this bit is set, the MAC transmitter and receiver also consider the S-VLAN (Type = 0x88A8) frames as valid VLAN tagged frames.	1'b0
[17]	R/W	VTIM	VLAN Tag Inverse Match Enable When set, this bit enables the VLAN Tag inverse matching. The frames that do not have matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The frames with matched VLAN Tag are marked as matched.	1'b0
[16]	R/W	ETV	Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. Similarly, when enabled, only 12 bits of the VLAN tag in the received frame are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN frame are used for comparison and VLAN hash filtering.	1'b0
[15:0]	R/W	VL	VLAN Tag Identifier for Receive Frames This field contains the 802.1Q VLAN tag to identify the VLAN frames and is compared to the 15th and 16th bytes of the frames being received for VLAN frames. The following list describes the bits of this field: <ul style="list-style-type: none"> ■ Bits [15:13]: User Priority ■ Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) ■ Bits[11:0]: VLAN tag's VLAN Identifier (VID) field When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the MAC does not check the fifteenth and 16th bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 or 0x88a8 as VLAN frames.	16'b0
Ethernet MAC Register 8 (Version Register)				
Address : C006_0020h				
[31:16]	-	RESERVED	Reserved	16'b0
[15:8]	R	USERVER	User-defined Version (configured with coreConsultant)	8'bx
[7:0]	R	SNPSVER	MAC Version (3.7)	8'h37

Bit	R/W	Symbol	Description	Reset Value
Ethernet MAC Register 9 (Debug Register)				
Address : C006_0024h				
[31:26]	-	RESERVED	Reserved	6'b0
[25]	R	TXSTSFSTS	<p>MTL TxStatus FIFO Full Status</p> <p>When high, this bit indicates that the MTL TxStatus FIFO is full. Therefore, the MTL cannot accept any more frames for transmission. This bit is reserved in the GMAC-AHB and GMAC-DMA configurations.</p>	1'b0
[24]	R	TXFSTS	<p>MTL Tx FIFO Not Empty Status</p> <p>When high, this bit indicates that the MTL Tx FIFO is not empty and some data is left for transmission.</p>	1'b0
[23]	-	RESERVED	Reserved	1'b0
[22]	R	TWCSTS	<p>MTL Tx FIFO Write Controller Status</p> <p>When high, this bit indicates that the MTL Tx FIFO Write Controller is active and is transferring data to the Tx FIFO.</p>	1'b0
[21:20]	R	TRCSTS	<p>MTL Tx FIFO Read Controller Status</p> <p>This field indicates the state of the Tx FIFO Read Controller:</p> <ul style="list-style-type: none"> ■ 00: IDLE state ■ 01: READ state (transferring data to the MAC transmitter) ■ 10: Waiting for TxStatus from the MAC transmitter ■ 11: Writing the received TxStatus or flushing the Tx FIFO 	2'b0
[19]	R	TXPAUSED	<p>MAC Transmitter in Pause</p> <p>When high, this bit indicates that the MAC transmitter is in the Pause condition (in the full-duplex-only mode) and hence does not schedule any frame for transmission.</p>	1'b0
[18:17]	R	TFCSTS	<p>MAC Transmit Frame Controller Status</p> <p>This field indicates the state of the MAC Transmit Frame Controller module:</p> <ul style="list-style-type: none"> ■ 00: IDLE state ■ 01: Waiting for status of previous frame or IFG or backoff period to be over ■ 10: Generating and transmitting a Pause frame (in the full-duplex mode) ■ 11: Transferring input frame for transmission 	2'b0
[16]	R	TPESTS	<p>MAC GMII or MII Transmit Protocol Engine Status</p> <p>When high, this bit indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data and is not in the IDLE state.</p>	1'b0
[15:0]	-	RESERVED	Reserved	16'b0
RESERVED				
Address : C006_0028h ~ C006_0034h				
Ethernet MAC Register 14 (Interrupt Status Register)				
Address : C006_0038h				
[31:12]	-	RESERVED	Reserved	20'b0
[11]	R	GPIIS	<p>GPI Interrupt Status</p> <p>When the GPIO feature is enabled, this bit is set when any active event (LL or LH) occurs on the GPIS field (Bits [3:0]) of Register 56 (General Purpose IO Register) and the corresponding GPIE bit is enabled. This bit is cleared on reading lane 0 (GPIS) of Register 56 (General Purpose IO Register). When the GPIO feature is not enabled, this bit is reserved.</p>	1'b0
[10]	R	LPIIS	<p>LPI Interrupt Status</p> <p>When the Energy Efficient Ethernet feature is enabled, this bit is set for any LPI state entry or exit</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
			in the MAC Transmitter or Receiver. This bit is cleared on reading Bit 0 of Register 12 (LPI Control and Status Register). In all other modes, this bit is reserved.	
[9]	R	TSIS	<p>Timestamp Interrupt Status</p> <p>When the Advanced Timestamp feature is enabled, this bit is set when any of the following conditions is true:</p> <ul style="list-style-type: none"> ■ The system time value equals or exceeds the value specified in the Target Time High and Low registers. ■ There is an overflow in the seconds register. ■ The Auxiliary snapshot trigger is asserted. <p>This bit is cleared on reading Bit 0 of Register 458 (Timestamp Status Register).</p> <p>If default Timestamping is enabled, when set, this bit indicates that the system time value is equal to or exceeds the value specified in the Target Time registers. In this mode, this bit is cleared after the completion of the read of this bit. In all other modes, this bit is reserved.</p>	1'b0
[8]	-	RESERVED	Reserved	1'b0
[7]	R	MMCRXIPIS	<p>MMC Receive Checksum Offload Interrupt Status</p> <p>This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.</p> <p>This bit is valid only when you select the optional MMC module and Checksum Offload Engine (Type 2) during core configuration.</p>	1'b0
[6]	R	MMCTXIS	<p>MMC Transmit Interrupt Status</p> <p>This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.</p> <p>This bit is valid only when you select the optional MMC module during core configuration.</p>	1'b0
[5]	R	MMCRXIS	<p>MMC Receive Interrupt Status</p> <p>This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.</p> <p>This bit is valid only when you select the optional MMC module during core configuration.</p>	1'b0
[4]	R	MMCIS	<p>MMC Interrupt Status</p> <p>This bit is set high when any of the Bits [7:5] is set high and cleared only when all of these bits are low.</p> <p>This bit is valid only when you select the optional MMC module during core configuration.</p>	1'b0
[3]	R	PMTIS	<p>PMT Interrupt Status</p> <p>This bit is set when a magic packet or remote wake-up frame is received in the power-down mode (see Bits 5 and 6 in the PMT Control and Status Register). This bit is cleared when both Bits[6:5] are cleared because of a read operation to the PMT Control and Status register.</p> <p>This bit is valid only when you select the optional PMT module during core configuration.</p>	1'b0
[2]	R	PCSANCIS	<p>PCS Auto-Negotiation Complete</p> <p>This bit is set when the Auto-negotiation is completed in the TBI, RTBI, or SGMII PHY interface (Bit 5 in Register 49 (AN Status Register)). This bit is cleared when you perform a read operation to the AN Status register.</p> <p>This bit is valid only when you select the optional TBI, RTBI, or SGMII PHY interface during core configuration and operation</p>	1'b0
[1]	R	PCSLCHGIS	<p>PCS Link Status Changed</p> <p>This bit is set because of any change in Link Status in the TBI, RTBI, or SGMII PHY interface (Bit 2 in Register 49 (AN Status Register)). This bit is cleared when you perform a read operation on the AN Status register.</p> <p>This bit is valid only when you select the optional TBI, RTBI, or SGMII PHY interface during core configuration and operation.</p>	1'b0
[0]	R	RGSMIIIS	RGMII or SMII Interrupt Status	1'b0

Bit	R/W	Symbol	Description	Reset Value
			This bit is set because of any change in value of the Link Status of RGMII or SMII interface (Bit 3 in Register 54 (SGMII/RGMII/SMII Control and Status Register)). This bit is cleared when you perform a read operation on the SGMII/RGMII/SMII Control and Status Register. This bit is valid only when you select the optional RGMII or SMII PHY interface during core configuration and operation.	
Ethernet MAC Register 15 (Interrupt Mask Register)				
Address : C006_003Ch				
[31:11]	-	RESERVED	Reserved	21'b0
[10]	R	LPIIM	LPI Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the LPI Interrupt Status bit in Register 14 (Interrupt Status Register). This bit is valid only when you select the Energy Efficient Ethernet feature during core configuration. In all other modes, this bit is reserved.	1'b0
[9]	R	TSIM	Timestamp Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of Timestamp Interrupt Status bit in Register 14 (Interrupt Status Register). This bit is valid only when IEEE1588 timestamping is enabled. In all other modes, this bit is reserved.	1'b0
[8:4]	-	RESERVED	Reserved	5'b0
[3]	R	PMTIM	PMT Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of PMT Interrupt Status bit in Register 14 (Interrupt Status Register).	1'b0
[2]	R	PCSANCIM	PCS AN Completion Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of PCS Auto-negotiation complete bit in Register 14 (Interrupt Status Register).	1'b0
[1]	R	PCSLCHGIM	PCS Link Status Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the PCS Link-status changed bit in Register 14 (Interrupt Status Register).	1'b0
[0]	R	RGSMMIIM	RGMII or SMII Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the RGMII or SMII Interrupt Status bit in Register 14 (Interrupt Status Register).	1'b0
Ethernet MAC Register 16 (MAC Address0 High Register)				
Address : C006_0040h				
[31]	R	AE	Address Enable This bit is always set to 1.	1'b0
[30:16]	-	RESERVED	Reserved	15'b0
[15:0]	R/W	ADDRHI	MAC Address0[47:32] This field contains the upper 16 bits (47:32) of the first 6-byte MAC address. The MAC uses this field for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.	16'hFFFF
Ethernet MAC Register 17 (MAC Address0 Low Register)				
Address : C006_0044h				
[31:0]	R/W	ADDRLO	MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. This is used by the MAC for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.	32'hFFFFFF FF

Bit	R/W	Symbol	Description	Reset Value
Ethernet MAC Register 18 (MAC Address1 High Register)				
Address : C006_0048h				
[31]	R	AE	Address Enable When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.	1'b0
[30]	R	SA	Source Address When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received frame.	1'b0
[29:24]	R	MBC	Mask Byte Control These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: <ul style="list-style-type: none">■ Bit 29: Register 18[15:8]■ Bit 28: Register 18[7:0]■ Bit 27: Register 19[31:24]■ ...■ Bit 24: Register 19[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.	6'b0
[23:16]	-	RESERVED	Reserved	8'b0
[15:0]	R/W	ADDRHI	MAC Address1 [47:32] This field contains the upper 16 bits (47:32) of the second 6-byte MAC address.	16'hFFFF
Ethernet MAC Register 18 (MAC Address1 Low Register)				
Address : C006_004Ch				
[31:0]	R/W	ADDRLO	MAC Address1 [31:0] This field contains the lower 32 bits of the second 6-byte MAC address. The content of this field is undefined until loaded by the Application after the initialization process.	32'hFFFFFF FF
RESERVED				
Address : C006_0050h ~ C006_00BCh				
Ethernet MAC Register 48 (AN Control Register)				
Address : C006_00C0h				
[31:19]	-	RESERVED	Reserved	13'b0
[18]	R/W	SGMRAL	SGMII RAL Control When set, this bit forces the SGMII RAL block to operate in the speed configured in the Speed and Port Select bits of the MAC Configuration register. This is useful when the SGMII interface is used in a direct MAC to MAC connection (without a PHY) and any MAC must reconfigure the speed. When reset, the SGMII RAL block operates according to the link speed status received on SGMII (from the PHY). This bit is reserved (and RO) if the SGMII PHY interface is not selected during core configuration.	1'b0
[17]	R/W	LR	Lock to Reference When set, this bit enables the PHY to lock its PLL to the 125 MHz reference clock. This bit controls the pcs_lck_ref_o signal on the TBI, RTBI, or SGMII interface.	1'b0
[16]	R/W	ECD	Enable Comma Detect	1'b0

Bit	R/W	Symbol	Description	Reset Value
			When set, this bit enables the PHY for comma detection and word resynchronization. This bit controls the pcs_en_cdet_o signal on the TBI, RTBI, or SGMII interface.	
[15]	-	RESERVED	Reserved	1'b0
[14]	R/W	ELE	External Loopback Enable When set, this bit causes the PHY to loopback the transmit data into the receive path. The pcs_ewrap_o signal is asserted high when this bit is set.	1'b0
[13]	-	RESERVED	Reserved	1'b0
[12]	R/W	ANE	External Loopback Enable When set, this bit causes the PHY to loopback the transmit data into the receive path. The pcs_ewrap_o signal is asserted high when this bit is set.	1'b0
[11:10]	-	RESERVED	Reserved	2'b0
[9]	R/W	RAN	Restart Auto-Negotiation When set, this bit causes auto-negotiation to restart if Bit 12 (ANE) is set. This bit is self-clearing after auto-negotiation starts. This bit should be cleared for normal operation.	1'b0
[8:0]	-	RESERVED	Reserved	9'b0

Ethernet MAC Register 49 (AN Status Register)

Address : C006_00C4h

[31:9]	-	RESERVED	Reserved	23'b0
[8]	R	SGMRAL	Extended Status This bit is tied to high if the TBI or RTBI interface is selected during core configuration indicating that the MAC supports extended status information in Register 53 (TBI Extended Status Register). This bit is tied to low if the SGMII interface is selected and the TBI or RTBI interface is not selected during core configuration indicating that Register 53 is not present.	1'b0
[7:6]	-	RESERVED	Reserved	2'b0
[5]	R	ANC	Auto-Negotiation Complete When set, this bit indicates that the auto-negotiation process is complete. This bit is cleared when auto-negotiation is reinitiated.	1'b0
[4]	-	RESERVED	Reserved	1'b0
[3]	R	ANA	Auto-Negotiation Ability This bit is always high because the MAC supports autonegotiation.	1'b0
[2]	R	LS	Link Status When set, this bit indicates that the link is up between the MAC and the TBI, RTBI, or SGMII interface. When cleared, this bit indicates that the link is down between the MAC and the TBI, RTBI, or SGMII interface.	1'b0
[1:0]	-	RESERVED	Reserved	2'b0

Ethernet MAC Register 50 (Auto-Negotiation Advertisement Register)

Address : C006_00C8h

[31:16]	-	RESERVED	Reserved	16'b0
[15]	R	NP	Next Page Support This bit is always low because the MAC does not support the next page.	1'b0
[14]	-	RESERVED	Reserved	1'b0
[13:12]	R/W	RFE	Remote Fault Encoding These bits provide a remote fault encoding, indicating to a link partner that a fault or error	2'b0

Bit	R/W	Symbol	Description	Reset Value
			condition has occurred. The encoding of these bits is defined in IEEE 802.3z, Section 37.2.1.5.	
[11:9]	-	RESERVED	Reserved	3'b0
[8:7]	R/W	PSE	Pause Encoding These bits provide an encoding for the Pause bits, indicating that the MAC is capable of configuring the Pause function as defined in IEEE 802.3x. The encoding of these bits is defined in IEEE 802.3z, Section 37.2.1.4.	2'b0
[6]	R/W	HD	Half-Duplex When set high, this bit indicates that the MAC supports the half-duplex mode. This bit is always low (and RO) when the MAC is configured for the full-duplex-only mode.	1'b0
[5]	R/W	FD	Full-Duplex When set high, this bit indicates that the MAC supports the full-duplex mode	1'b0
[4:0]	-	RESERVED	Reserved	5'b0

Ethernet MAC Register 51 (Auto-Negotiation Link Partner Ability Register)

Address : C006_00CCh

[31:16]	-	RESERVED	Reserved	16'b0
[15]	R	NO	Next Page Support When set, this bit indicates that more next page information is available. When cleared, this bit indicates that next page exchange is not desired.	1'b0
[14]	R	ACK	Acknowledge When set, the auto-negotiation function uses this bit to indicate that the link partner has successfully received the base page of the MAC. When cleared, it indicates that the link partner did not successfully receive the base page of the MAC.	1'b0
[13:12]	R	RFE	Remote Fault Encoding These bits provide a remote fault encoding, indicating a fault or error condition of the link partner. The encoding of these bits is defined in IEEE 802.3z, Section 37.2.1.5.	2'b0
[11:9]	-	RESERVED	Reserved	3'b0
[8:7]	R	PSE	Pause Encoding These bits provide an encoding for the Pause bits, indicating that the link partner's capability of configuring the Pause function as defined in the IEEE 802.3x specification. The encoding of these bits is defined in IEEE 802.3z, Section 37.2.1.4.	2'b0
[6]	R	HD	Half-Duplex When set, this bit indicates that the link partner has the ability to operate in the half-duplex mode. When cleared, this bit indicates that the link partner does not have the ability to operate in the half-duplex mode.	1'b0
[5]	R	FD	Full-Duplex When set, this bit indicates that the link partner has the ability to operate in the fullduplex mode. When cleared, this bit indicates that the link partner does not have the ability to operate in the full-duplex mode.	1'b0
[4:0]	-	RESERVED	Reserved	5'b0

Ethernet MAC Register 52 (Auto-Negotiation Expansion Register)

Address : C006_00D0h

[31:3]	-	RESERVED	Reserved	29'b0
[2]	R	NPA	Next Page Ability This bit is always low because the MAC does not support the next page function.	1'b0
[1]	R	NPR	New Page Received	1'b0

Bit	R/W	Symbol	Description	Reset Value		
			When set, this bit indicates that the MAC has received a new page. This bit is cleared when read.			
[0]	-	RESERVED	Reserved	1'b0		
Ethernet MAC Register 53 (TBI Extended Status Register)						
Address : C006_00D4h						
[31:16]	-	RESERVED	Reserved	29'b0		
[15]	R	GFD	1000BASE-X Full-Duplex Capable This bit indicates that the MAC is able to perform the full-duplex and 1000BASE-X operations.	1'b0		
[14]	R	GHD	1000BASE-X Half-Duplex Capable This bit indicates that the MAC is able to perform the half-duplex and 1000BASE-X operations. This bit is always low when the MAC is configured for the full-duplex-only operation during core configuration.	1'b0		
[13:0]	-	RESERVED	Reserved	29'b0		
Ethernet MAC Register 54 (SGMII/RGMII/SMII Control and Status Register)						
Address : C006_00D8h						
[31:17]	-	RESERVED	Reserved	15'b0		
[16]	R/W	SMIDRXS	Delay SMII RX Data Sampling with respect to the SMII SYNC Signal When set, the first bit of the SMII RX data is sampled one cycle after the SMII SYNC signal. When reset, the first bit of the SMII RX data is sampled along with the SMII SYNC signal. If the SMII PHY Interface with source synchronous mode is selected during core configuration, this bit is reserved (RO with default value).	1'b0		
[15:6]	-	RESERVED	Reserved	10'b0		
[5]	R	FALSCARDET	False Carrier Detected This bit indicates whether the SMII PHY detected false carrier (1'b1). This bit is reserved when the MAC is configured for the SGMII or RGMII PHY interface.	1'b0		
[4]	R	JABTO	Jabber Timeout This bit indicates whether there is jabber timeout error (1'b1) in the received frame. This bit is reserved when the MAC is configured for the SGMII or RGMII PHY interface.	1'b0		
[3]	R	LNKSTS	Link Status When set, this bit indicates that the link is up between the local PHY and the remote PHY. When cleared, this bit indicates that the link is down between the local PHY and the remote PHY.	1'b0		
[2:1]	R	LNKSPEED	Link Speed This bit indicates the current speed of the link: ■ 00: 2.5 MHz ■ 01: 25 MHz ■ 10: 125 MHz Bit 2 is reserved when the MAC is configured for the SMII PHY interface.	1'b0		
[0]	R	LNKMOD	Link Mode This bit indicates the current mode of operation of the link: ■ 1'b0: Half-duplex mode ■ 1'b1: Full-duplex mode	1'b0		
RESERVED						
Address : C006_00E0h ~ C006_06FCh						
Ethernet MAC Register 448 (Time Stamp Control Register)						
Address : C006_0700h						

Bit	R/W	Symbol	Description	Reset Value
[31:29]	-	RESERVED	Reserved	3'b0
[28]	R/W	ATSEN3	<p>Auxiliary Snapshot 3 Enable</p> <p>This field controls capturing the Auxiliary Snapshot Trigger 3. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[3] input is enabled. When this bit is reset, the events on this input are ignored.</p> <p>This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration or the selected number in the Number of IEEE 1588 Auxiliary Snapshot Inputs option is less than four.</p>	1'b0
[27]	R/W	ATSEN2	<p>Auxiliary Snapshot 2 Enable</p> <p>This field controls capturing the Auxiliary Snapshot Trigger 2. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[2] input is enabled. When this bit is reset, the events on this input are ignored.</p> <p>This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration or the selected number in the Number of IEEE 1588 Auxiliary Snapshot Inputs option is less than three.</p>	16'b0
[26]	R/W	ATSEN1	<p>Auxiliary Snapshot 1 Enable</p> <p>This field controls capturing the Auxiliary Snapshot Trigger 1. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[1] input is enabled. When this bit is reset, the events on this input are ignored.</p> <p>This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration or the selected number in the Number of IEEE 1588 Auxiliary Snapshot Inputs option is less than two.</p>	1'b0
[25]	R/W	ATSEN0	<p>Auxiliary Snapshot 0 Enable</p> <p>This field controls capturing the Auxiliary Snapshot Trigger 0. When this bit is set, the Auxiliary snapshot of event on ptp_aux_trig_i[0] input</p> <p>is enabled. When this bit is reset, the events on this input are ignored. This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration.</p>	1'b0
[24]	R/W	ATSFC	<p>Auxiliary Snapshot FIFO Clear</p> <p>When set, it resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, auxiliary snapshots get stored in the FIFO. This bit is reserved when the Add IEEE 1588 Auxiliary Snapshot option is not selected during core configuration.</p>	1'b0
[23:19]	-	RESERVED	Reserved	3'b0
[18]	R/W	TSENMACADDR	<p>Enable MAC address for PTP Frame Filtering</p> <p>When set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP frames when PTP is directly sent over Ethernet.</p>	1'b0
[17:16]	R/W	SNAPTYPSEL	<p>Select PTP packets for Taking Snapshots</p> <p>These bits along with Bits 15 and 14 decide the set of PTP packet types for which snapshot needs to be taken.</p>	2'b0
[15]	R/W	TSMSTRENA	<p>Enable Snapshot for Messages Relevant to Master</p> <p>When set, the snapshot is taken only for the messages relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node.</p>	1'b0
[14]	R/W	TSEVNTEA	<p>Enable Timestamp Snapshot for Event Messages</p> <p>When set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When reset, the snapshot is taken for all messages except Announce, Management, and Signaling.</p>	1'b0
[13]	R/W	TSIPV4ENA	<p>Enable Processing of PTP Frames Sent over IPv4-UDP</p> <p>When set, the MAC receiver processes the PTP packets encapsulated in UDP over IPv4 packets. When this bit is clear, the MAC ignores the PTP transported over UDP-IPv4 packets. This bit is set by default.</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
[12]	R/W	TSIPV6ENA	Enable Processing of PTP Frames Sent over IPv6-UDP When set, the MAC receiver processes PTP packets encapsulated in UDP over IPv6 packets. When this bit is clear, the MAC ignores the PTP transported over UDP-IPv6 packets.	1'b0
[11]	R/W	TSIPENA	Enable Processing of PTP over Ethernet Frames When set, the MAC receiver processes the PTP packets encapsulated directly in the Ethernet frames. When this bit is clear, the MAC ignores the PTP over Ethernet packets.	1'b0
[10]	R/W	TSVER2ENA	Enable PTP packet Processing for Version 2 Format When set, the PTP packets are processed using the 1588 version 2 format. Otherwise, the PTP packets are processed using the version 1 format.	1'b0
[9]	R/W	TSCTRLSSR	Timestamp Digital or Binary Rollover Control When set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (High) seconds. When reset, the rollover value of sub-second register is 0x7FFF_FFFF. The sub-second increment has to be programmed correctly depending on the PTP reference clock frequency and the value of this bit.	1'b0
[8]	R/W	TSENALL	Enable Timestamp for All Frames When set, the timestamp snapshot is enabled for all frames received by the MAC.	1'b0
[7:6]	-	RESERVED	Reserved	2'b0
[5]	R/W	TSADDREG	Addend Reg Update When set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This is cleared when the update is completed. This register bit should be zero before setting it.	1'b0
[4]	R/W	TSTRIG	Timestamp Interrupt Trigger Enable When set, the timestamp interrupt is generated when the System Time becomes greater than the value written in the Target Time register. This bit is reset after the generation of the Timestamp Trigger Interrupt.	1'b0
[3]	R/W	TSUPDT	Timestamp Update When set, the system time is updated (added or subtracted) with the value specified in Register 452 (System Time – Seconds Update Register) and Register 453 (System Time – Nanoseconds Update Register). This bit should be read zero before updating it. This bit is reset when the update is completed in hardware. The "Timestamp Higher Word" register (if enabled during core configuration) is not updated.	1'b0
[2]	R/W	TSINIT	Timestamp Initialize When set, the system time is initialized (overwritten) with the value specified in the Register 452 (System Time – Seconds Update Register) and Register 453 (System Time – Nanoseconds Update Register). This bit should be read zero before updating it. This bit is reset when the initialization is complete. The "Timestamp Higher Word" register (if enabled during core configuration) can only be initialized.	1'b0
[1]	R/W	TSCFUPDT	Timestamp Fine or Coarse Update When set, this bit indicates that the system times update should be done using the fine update method. When reset, it indicates the system timestamp update should be done using the Coarse method.	1'b0
[0]	R/W	TSENA	Timestamp Enable When set, the timestamp is added for the transmit and receive frames. When disabled, timestamp is not added for the transmit and receive frames and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the receive side, the MAC processes the 1588 frames only if this bit is set.	1'b0
Ethernet MAC Register 449 (Sub-Second Increment Register)				

Bit	R/W	Symbol	Description	Reset Value
Address : C006_0704h				
[31:8]	-	RESERVED	Reserved	24'b0
[7:0]	R/W	SSINC	<p>Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of clk_ptp_i) with the contents of the sub-second register. For example, when PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time- Nanoseconds register has an accuracy of 1 ns [Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register)]. When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465ns. In this case, you should program a value of 43 (0x2B) that is derived by 20ns/0.465.</p>	8'b0
Ethernet MAC Register 450 (System Time - Second Register)				
Address : C006_0708h				
[31:0]	R	TSS	<p>Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the MAC.</p>	32'b0
Ethernet MAC Register 451 (System Time – Nano second Register)				
Address : C006_070Ch				
[31]	-	RESERVED	Reserved	1'b0
[30:0]	R	TSSS	<p>Timestamp Sub Seconds The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register), each bit represents 1 ns and the maximum value is 0xB9A_C9FF, after which it rolls-over to zero.</p>	31'b0
Ethernet MAC Register 452 (System Time – Second Update Register)				
Address : C006_0710h				
[31:0]	R/W	TSS	<p>Timestamp Second The value in this field indicates the time in seconds to be initialized or added to the system time.</p>	32'b0
Ethernet MAC Register 453 (System Time – Nano Second Update Register)				
Address : C006_0714h				
[31]	R/W	ADDSUB	<p>Add or Subtract Time When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register.</p>	1'b0
[30:0]	R/W	TSSS	<p>Timestamp Sub Seconds The value in this field has the sub second representation of time, with an accuracy of 0.46 ns. When Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register), each bit represents 1 ns and the programmed value should not exceed 0xB9A_C9FF.</p>	31'b0
Ethernet MAC Register 454 (Timestamp Addend Register)				
Address : C006_0718h				
[31:0]	R/W	TSAR	<p>Timestamp Addend Register This field indicates the 32-bit time value to be added to the Accumulator register to achieve time synchronization..</p>	32'b0
Ethernet MAC Register 455 (Target Time Second Register)				
Address : C006_071Ch				
[31:0]	R/W	TSTR	<p>Target Time Seconds Register This register stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, then based on Bits [6:5] of Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled).</p>	32'b0
Ethernet MAC Register 456 (Target Time Nano Second Register)				

Bit	R/W	Symbol	Description	Reset Value
Address : C006_0720h				
[31]	R/W	TRGTBUSY	<p>Target Time Register Busy</p> <p>The MAC sets this bit when the PPSCMD field (Bit [3:0]) in Register 459 (PPS Control Register) is programmed to 010 or 011. Programming the PPSCMD field to 010 or 011, instructs the MAC to synchronize the Target Time Registers to the PTP clock domain.</p> <p>The MAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain. The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previous programmed time gets corrupted. This bit is reserved when the Enable Flexible Pulse-Per-Second Output feature is not selected.</p>	1'b0
[30:0]	R/W	TTSLO	<p>Target Timestamp Low Register</p> <p>This register stores the time in (signed) nanoseconds. When the value of the timestamp matches the both Target Timestamp registers, then based on the TRGTMODSEL0 field (Bits [6:5]) in Register 459 (PPS Control Register), the MAC starts or stops the PPS signal output and generates an interrupt (if enabled).</p> <p>This value should not exceed 0x3B9A_C9FF when Bit 9 (TSCTRLSSR) is set in Register 448 (Timestamp Control Register). The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value.</p>	31'b0
Ethernet MAC Register 457 (System Time – Higher Word Seconds Register)				
Address : C006_0724h				
[31:16]	-	RESERVED	Reserved	16'b0
[15:0]	R/W	TSHWR	<p>Timestamp Higher Word Register</p> <p>This field contains the most significant 16-bits of the timestamp seconds value. This register is optional and can be selected using the Enable IEEE 1588 Higher Word Register option during core configuration. The register is directly written to initialize the value. This register is incremented when there is an overflow from the 32-bits of the System Time – Seconds register.</p>	16'b0
Ethernet MAC Register 458 (Timestamp Status Register)				
Address : C006_0728h				
[31:30]	-	RESERVED	Reserved	2'b0
[29:25]	R	ATSNS	<p>Number of Auxiliary Timestamp Snapshots</p> <p>This field indicates the number of Snapshots available in the FIFO. A value equal to the selected depth of FIFO (4, 8, or 16) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 0000) when the Auxiliary snapshot FIFO clear bit is set. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected during core configuration.</p>	5'b0
[24]	R	ATSSTM	<p>Auxiliary Timestamp Snapshot Trigger Missed</p> <p>This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO. This bit is valid only if the Add IEEE 1588 Auxiliary Snapshot option is selected during core configuration.</p>	1'b0
[23:20]	-	RESERVED	Reserved	4'b0
[19:16]	R	ATSSTN	<p>Auxiliary Timestamp Snapshot Trigger Identifier</p> <p>These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one.</p> <p>One bit is assigned for each trigger as shown in the following list:</p> <ul style="list-style-type: none"> ■ Bit 16: Auxiliary trigger 0 ■ Bit 17: Auxiliary trigger 1 ■ Bit 18: Auxiliary trigger 2 	4'b0

Bit	R/W	Symbol	Description	Reset Value
			■ Bit 19: Auxiliary trigger 3 The software can read this register to find the triggers that are set when the timestamp is taken.	
[15:10]	-	RESERVED	Reserved	6'b0
[9]	R	TSTRGTERR3	Timestamp Target Time Error This bit is set when the target time, being programmed in Register 496 and Register 497, is already elapsed. This bit is cleared when read by the application.	1'b0
[8]	R	TSTARTT3	Timestamp Target Time Reached for Target Time PPS3 When set, this bit indicates that the value of system time is greater than or equal to the value specified in Register 496 (PPS3 Target Time High Register) and Register 497 (PPS3 Target Time Low Register).	1'b0
[7]	R	TSTRGTERR2	Timestamp Target Time Error This bit is set when the target time, being programmed in Register 488 and Register 489, is already elapsed. This bit is cleared when read by the application.	1'b0
[6]	R	TSTARTT2	Timestamp Target Time Reached for Target Time PPS2 When set, this bit indicates that the value of system time is greater than or equal to the value specified in Register 488 (PPS2 Target Time High Register) and Register 489 (PPS2 Target Time Low Register).	1'b0
[5]	R	TSTRGTERR1	Timestamp Target Time Error This bit is set when the target time, being programmed in Register 480 and Register 481, is already elapsed. This bit is cleared when read by the application.	1'b0
[4]	R	TSTARTT1	Timestamp Target Time Reached for Target Time PPS1 When set, this bit indicates that the value of system time is greater than or equal to the value specified in Register 480 (PPS1 Target Time High Register) and Register 481 (PPS1 Target Time Low Register).	1'b0
[3]	R	TSTRGTERR	Timestamp Target Time Error This bit is set when the target time, being programmed in Register 455 and Register 456, is already elapsed. This bit is cleared when read by the application.	1'b0
[2]	R	AUXTSTRIG	Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO. This bit is valid only if the Enable IEEE 1588 Auxiliary Snapshot feature is selected.	1'b0
[1]	R	TSTARTT	Timestamp Target Time Reached When set, this bit indicates that the value of system time is greater than or equal to the value specified in the Register 455 (Target Time Seconds Register) and Register 456 (Target Time Nanoseconds Register).	1'b0
[0]	R	TSSOVF	Timestamp Seconds Overflow When set, this bit indicates that the seconds value of the timestamp (when supporting version 2 format) has overflowed beyond 32'hFFFF_FFFF.	1'b0

Section 17. SD/MMC Controller

17.1 Overview

This Section describes Secure Digital (SD/SDIO), MultiMediaCard (MMC), CT-ATA host controller and related register that NXP4330D/Q supports. The Mobile Storage Host is an interface between system and SD/MMC card.

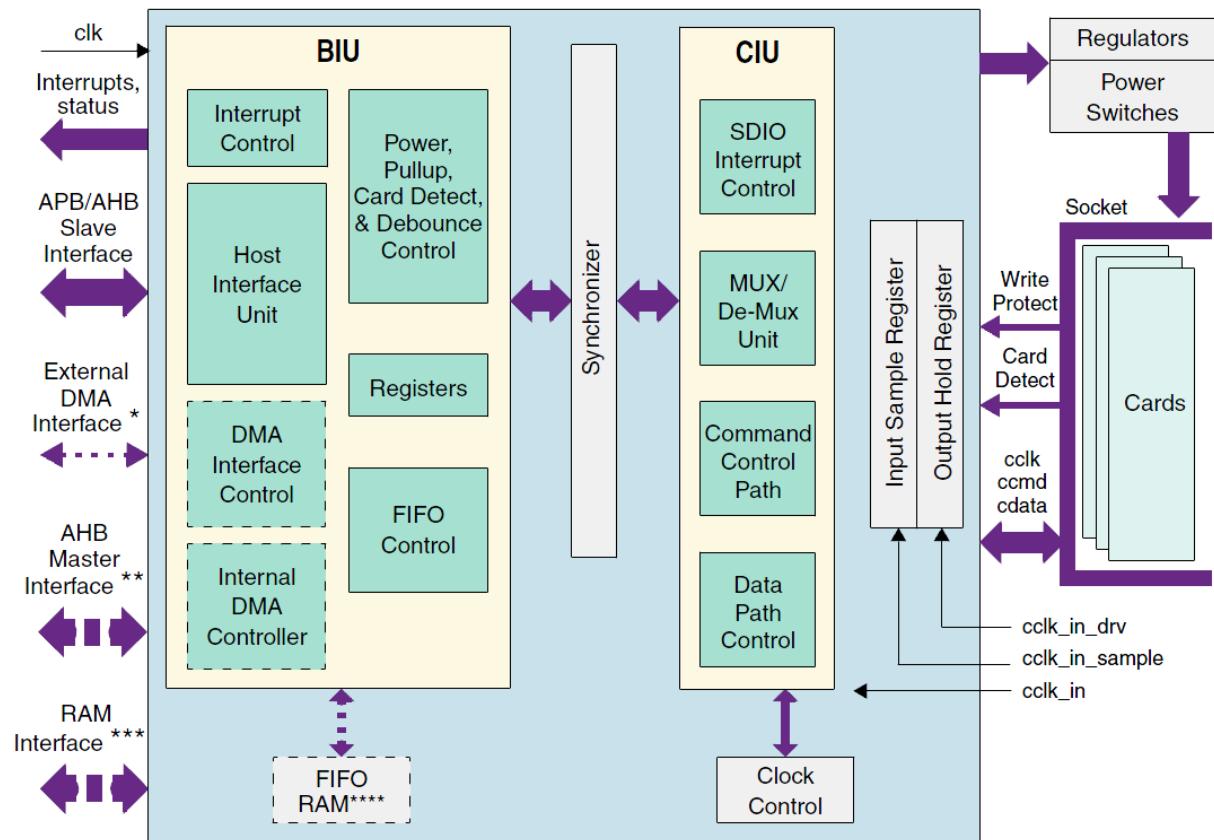
17.1.1 Features

- Compatible with the Multi-Media Card System Specification, (MMC 4.41, eMMC 4.5)
- Compatible with the Secure Digital memory Specification, (SD 3.0)
- Compatible with the Secure Digital I/O Specification, (SDIO 3.0)
- Supports clock speeds up to 50 MHz
- Contains an Internal Clock Pre-Scaler
- Contains 32 Bytes of FIFO for data Receive/Transmit
- 3 Channel of SD/MMC

17.1.1.1 Features of Mobile Storage Host

- The following are features of the Mobile Storage Host :
 - Supports Secure Digital memory protocol commands
 - Supports Secure Digital I/O protocol commands
 - Supports Multimedia Card protocol commands
 - Supports CE-ATA digital protocol commands
 - Supports Command Completion signal and interrupt to host processor
 - Command Completion Signal disable feature
- The following features of MMC4.41 are supported:
 - DDR in 4-bit mode
 - GO_PRE_IDLE_STATE command (CMD with argument 0xF0F0F0F0)
 - New EXTCSD registers
 - Hardware Reset as supported by eMMC 4.41
- The following IP-specific features of eMMC 4.5 are supported:
 - HS200 Mode (4 bits)
 - Packed Commands, CMD21, CMD49
 - Support for 1.2/1.8/3.3V of operation control
 - START bit behavior change for DDR modes
- The following IP-specific features of MMC 4.41 are not supported:
 - Boot in DDR mode

17.1.2 Block Diagram of Mobile Storage Host



* Optional External DMA Interface; present only if internal DMAC is *not* present

** Optional AHB/APB Master Interface; present only if internal DMAC is present

*** Optional RAM Interface

**** FIFO RAM can be chosen as either internal or external RAM

- optional

Note: The card_detect and write-protect signals are from the SD/MMC card socket and not from the SD/MMC card.

Figure 17-1 Block diagram of Mobile Storage Host

Figure 17-1 illustrates the block diagram of Mobile Storage Host.

- Bus Inferface Unit (BIU) : Provides AMBA AHB/APB and DMA interfaces for register and data read/writes
- Card Interface Unit (CIU) : Takes care of SD_MMC_CEATA protocols and provides clock management.

The BIU provides the host interface to the registers. It also provides the data FIFO through the Host Interface Unit (HIU). Additionally, it provides independent data FIFO access through a DMA interface. You can configure host interface as an AMBA APB slave interface.

The IDMAC is responsible for exchanging data between FIFO and the host memory. A set of IDMAC registers is accessible by host for controlling the IDMAC operation through the AMBA APB slave interface.

The Mobile Storage CIU controls the card-specific protocols. Within CIU, the command path control unit and data path control unit interface with the controller to the command and data ports of the SD_MMC_CEATA cards. The CIU also provides clock control.

17.1.3 Clock Phase Shifter

SD/MMC card receives DATA/CMD with card clock from the host controller. To synchronize the clock and DATA/CMD, it is required that the clock delay is inserted to the Tx/Rx clock path. For this to happen, the logic is added to the design as illustrated in the figure 17-2. And clock selection can be done with CLKSEL register at the end of this Section.

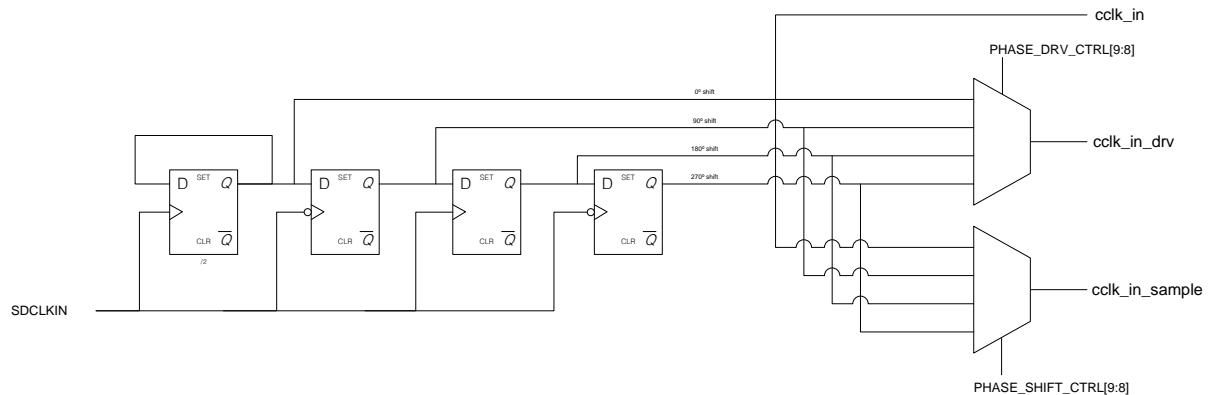


Figure 17-2 Block diagram of Mobile Storage Host

This clock phase shifter makes 0, 90, 180, 270 phase shifted clocks for Tx/Rx respectively. To make 50 MHz phase shifted clock, SDCLKIN should be 100 MHz.

17.2 Register Summary

Bit	R/W	Symbol	Description	Reset Value
			Bit automatically clears once response is sent. To wait for MMC card interrupts, host issues CMD40, and DWC_mobile_storage waits for interrupt response from MMC card(s). In meantime, if host wants DWC_mobile_storage to exit waiting for interrupt state, it can set this bit, at which time DWC_mobile_storage command state-machine sends CMD40 response on bus and returns to idle state.	
[6]	R/W	READ_WAIT	For sending read-wait to SDIO cards. 0 : Clear read wait 1 : Assert read wait	1'b0
[5]	-	REVERSED	Reserved	
[4]	R/W	INT_ENABLE	Global interrupt enable/disable bit 0 : Disable interrupt 1 : Enable interrupt The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.	1'b0
[3]	R/W	RESERVED	Reserved.	1'b0
[2]	R/W	DMA_RESET	0 : No change 1 : Reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared after two AHB clocks.	1'b0
[1]	R/W	FIFO_RESET	0 : No change 1 : Reset to data FIFO To reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation.	1'b0
[0]	R/W	CONTROLLER_RESET	0 – No change 1 – Reset DWC_mobile_storage controller To reset controller, firmware should set bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles. This resets: * BIU/CIU interface * CIU and state machines * abort_read_data, send_irq_response, and read_wait bits of Control register * start_cmd bit of Command register Does not affect any registers or DMA interface, or FIFO or hos interrupts	1'b0

SD/MMC Power Enable Register

Address : SD0 : C006_2004h SD1: C006_8004h SD2 : C006_9004h

[31:1]	-	RESERVED	Reserved	
[0]	R/W	POWER_ENABLE	Power on/off switch for up to 16 cards; for example, bit[0] controls card 0. Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card. 0 : power off 1 : power on Only NUM_CARDS number of bits are implemented. Bit values output to card_power_en port. Optional feature; ports can be used as general-purpose outputs.	1'b0

SD/MMC Clock Divider Register(CLKDIV)

Address : SD0 : C006_2008h SD1: C006_8008h SD2 : C006_9008h

[31:24]	R/W	CLK_DIVIDER3	Clock divider-3 value. Clock division is 2^n . For example, value of 0 means divide by $2^0 = 0$ (no division, bypass), a value of 1 means divide by $2^1 = 2$, a value of "ff" means divide by $2^{255} = 510$, and so on.	8'b0
[23:16]	R/W	CLK_DIVIDER2	Clock divider-2 value. Clock division is 2^n . For example, value of 0 means divide by $2^0 = 0$ (no division, bypass), a value of 1 means divide by $2^1 = 2$, a value of "ff" means divide by $2^{255} = 510$, and so on.	8'b0
[15:8]	R/W	CLK_DIVIDER1	Clock divider-1 value. Clock division is 2^n . For example, value of 0 means divide by $2^0 = 0$ (no division, bypass), a value of 1 means divide by $2^1 = 2$, a value of "ff" means divide by $2^{255} = 510$.	8'b0

Bit	R/W	Symbol	Description	Reset Value
			510, and so on.	
[7:0]	R/W	CLK_DIVIDER0	Clock divider-0 value. Clock division is 2^n . For example, value of 0 means divide by $2^0 = 0$ (no division, bypass), a value of 1 means divide by $2^1 = 2$, a value of "ff" means divide by $2^{255} = 510$, and so on.	8'b0
SD/MMC Clock Source Register				
Address : SD0 : C006_200Ch SD1: C006_800Ch SD2 : C006_900Ch				
[31:2]	-	RESERVED	Reserved	
[1:0]	R/W	CLK_SOURCE	Clock divider source 00 : Clock divider 0 10 : Clock divider 2	32'b0 01 : Clock divider 1 11 : Clock divider 3
SD/MMC Clock Enable Register (CLKENA)				
Address : SD0 : C006_2010h SD1: C006_8010h SD2 : C006_9010h				
[31:17]	-	RESERVED	Reserved	15'b0
[16]	R/W	CCLK_LOW_POWER	Low power control. If enabled, stop clock when card in idle status. 0 : Low power disable 1 : Low power enable	1'b0
[15:1]	R/W	RESERVED	Reserved	15'b0
[0]	R/W	CCLK_ENABLE	Clock enable control 0 : Disable sd clock 1 : Enable sd clock	1'b0
SD/MMC Timeout Register (TMOUT)				
Address : SD0 : C006_2014h SD1: C006_8014h SD2 : C006_9014h				
[31:8]	R/W	DATA_TIMEOUT	Value for card Data Read Timeout; same value also used for Data Starvation by Host timeout. The timeout counter is started only after the card clock is stopped. Value is in number of card output clocks – cclk_out of selected card.	24'hFF_FFF_F
[7:0]	R/W	RESPONSE_TIMEOUT	Response timeout value. Value is in number of card output clocks – cclk_out.	8'h40
SD/MMC Card Type Register (CTYPE)				
Address : SD0 : C006_2018h SD1: C006_8018h SD2 : C006_9018h				
[31:17]	-	RESERVED	Reserved	15'b0
[16]	R/W	CARD_WIDTH	8 bit interface mode. In NXP4330D/Q, this bit must be 0.	1'b0
[15:1]	-	RESERVED	Reserved.	15'b0
[0]	R/W	CARD_WIDTH	Card bus width. 0 : 1-bit mode 1 : 4-bit mode	1'b0
SDMMC Block Size Register (BLKSIZ)				
Address : SD0 : C006_201Ch SD1: C006_801Ch SD2 : C006_901Ch				
[31:16]	-	RESERVED	Reserved	16'b0
[15:0]	R/W	BLKSIZE	Block size in bytes.	16'h200
SD/MMC Byte Count Register (BYTCNT)				
Address : SD0 : C006_2020h SD1: C006_8020h SD2 : C006_9020h				
[31:0]	R/W	BYTE_COUNT	Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.	32'h200

Bit	R/W	Symbol	Description	Reset Value
SD/MMC Interrupt Mask Register (INTMASK)				
Address : SD0 : C006_2024h SD1: C006_8024h SD2 : C006_9024h				
[31:17]	-	RESERVED	Reserved	15'b0
[16]	R/W	SDIO_INT_MASK	SDIO interrupt mask. 0 : Masked 1 : Enabled	1'b0
[15]	R/W	MASK_EBE	End bit error (read), Write no CRC (write) interrupt mask.	1'b0
[14]	R/W	MASK_ACD	Auto command done interrupt mask.	1'b0
[13]	R/W	MASK_SBE	Start bit error / Busy Complete interrupt mask	1'b0
[12]	R/W	MASK_HLE	Hardware locked write error interrupt mask	1'b0
[11]	R/W	MASK_FRUN	FIFO underrun/overrun error interrupt mask	1'b0
[10]	R/W	MASK_HTO	Data starvation by host timeout / Volt_switch_int interrupt mask	1'b0
[9]	R/W	MASK_DRTO	Data read timeout interrupt mask	1'b0
[8]	R/W	MASK_RTO	Response timeout interrupt mask	1'b0
[7]	R/W	MASK_DCRC	Data CRC error interrupt mask	1'b0
[6]	R/W	MASK_RCRC	Response CRC error interrupt mask	1'b0
[5]	R/W	MASK_RXDR	Receive FIFO data request interrupt mask	1'b0
[4]	R/W	MASK_TXDR	Transmit FIFO data request interrupt mask	1'b0
[3]	R/W	MASK.DTO	Data transfer over interrupt mask	1'b0
[2]	R/W	MASK_CD	Command done interrupt mask	1'b0
[1]	R/W	MASK_RE	Response error interrupt mask	1'b0
[0]	W	MASK_CD	This bit must be '0'	1'b0
SD/MMC Command Argument Register (CMDARG)				
Address : SD0 : C006_2028h SD1: C006_8028h SD2 : C006_9028h				
[31:0]	R/W	CMDARG	Value indicates command argument to be passed to card.	32'b0
SD/MMC Command Register (CMD)				
Address : SD0 : C006_202Ch SD1: C006_802Ch SD2 : C006_902Ch				
[31]	R/W	START_CMD	Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD_MMC_CEATA cards, Command Done bit is set in raw interrupt register.	1'b0
[30]	-	RESERVED	Reserved	1'b0
[29]	R/W	USE_HOLD_REG	Use Hold Register 0 : CMD and DATA sent to card bypassing HOLD Register 1 : CMD and DATA sent to card through the HOLD Register	1'b1
[28]	R/W	VOLT_SWITCH	Voltage switch bit 0 : No voltage switching 1 : Voltage switching enabled; must be set for CMD11 only	1'b0
[27]	R/W	BOOT_MODE	Boot Mode	1'b0

Bit	R/W	Symbol	Description	Reset Value
			0 : Mandatory Boot operation 1 : Alternate Boot operation	
[26]	R/W	DISABLE_BOOT	Disable Boot. When software sets this bit along with start_cmd, CIU terminates the boot operation. Do NOT set disable_boot and enable_boot together.	1'b0
[25]	R/W	EXPECT_BOOT_ACK	Expect Boot Acknowledge. When Software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card.	1'b0
[24]	R/W	ENABLE_BOOT	Enable Boot—this bit should be set only for mandatory boot mode. When Software sets this bit along with start_cmd, CIU starts the boot sequence for the corresponding card by asserting the CMD line low. Do NOT set disable_boot and enable_boot together	1'b0
[23]	R/W	CCS_EXPECTED	0 : Interrupts are not enabled in CE-ATA device (nIEN = 1 in ATA control register), or command does not expect CCS from device 1 : Interrupts are enabled in CE-ATA device (nIEN = 0), and RW_BLK command expects command completion signal from CE-ATA device If the command expects Command Completion Signal (CCS) from the CE-ATA device, the software should set this control bit. DWC_mobile_storage sets Data Transfer Over (DTO) bit in RINTSTS register and generates interrupt to host if Data Transfer Over interrupt is not masked.	1'b0
[22]	R/W	READ_CEATA_DEVICE	0 : Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device 1 : Host is performing read access (RW_REG or RW_BLK) towards CE-ATA device Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. DWC_mobile_storage should not indicate read data timeout while waiting for data from CE-ATA device.	1'b0
[21]	R/W	UPDATE_CLOCK_REGISTER_ONLY	0 : Normal command sequence 1 : Do not send commands, just update clock register value into card clock domain Following register values transferred into card clock domain: CLKDIV, CLRSRC, CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_registers_only = 0, following control registers are transferred from BIU to CIU: CMD, CMDARG, TMOUT, CTYPE, BLKSIZ, BYTCNT. CIU uses new register values for new command sequence to card(s). When bit is set, there are no Command Done interrupts because no command is sent to SD_MMCEATA cards.	1'b0
[20:16]	R/W	CARD_NUMBER	Card number in use. In NXP4330D/Q, this value must be 0.	5'b0
[15]	R/W	SEND_INITIALIZATION	After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card. This bit should not be set for either of the boot modes (alternate or mandatory). 0 : Do not send initialization sequence before sending command 1 : Send initialization sequence before sending command	1'b0
[14]	R/W	STOP_ABORT_CMD	0 : Neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1 : Stop or abort command intended to stop current data transfer in progress. When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with CMD[26]=disable_boot.	1'b0
[13]	R/W	WAIT_PRVDATA_COMPLETE	0 : Send command at once, even if previous data transfer has not completed 1 : Wait for previous data transfer completion before sending command The wait_prvdata_complete = 0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command.	1'b0

Bit	R/W	Symbol	Description	Reset Value
[12]	R/W	SEND_AUTO_STOP	0 : No stop command sent at end of data transfer 1 : Send stop command at end of data transfer When set, mobile_storage sends stop command to SD_MMC_CEATA cards at end of data transfer * when send_auto_stop bit should be set, since some data transfers do not need explicit stop commands * open-ended transfers that software should explicitly send to stop command Additionally, when “resume” is sent to resume – suspended memory access of SD-Combo card – bit should be set correctly if suspended data transfer needs send_auto_stop. Don't care if no data expected from card.	1'b0
[11]	R/W	TRANSFER_MODE	Transfer mode. Don't care if no data expected. 0 : Block data transfer mode 1 : Stream data transfer mode	1'b0
[10]	R/W	READ/WRITE	Read/Write mode selection. Don't care if no data expected. 0 : Read from card 1 : Write to card	1'b0
[9]	R/W	DATA_EXPECTED	Data transfer expected flag. 0 : No data transfer expected 1 : Data transfer expected	1'b0
[8]	R/W	CHECK_RESPONSE_CRC	0 : Do not check response CRC 1 : Check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller.	1'b0
[7]	R/W	RESPONSE_LENGTH	Response length selection. 0 : Short response 1 : Long response	1'b0
[6]	R/W	RESPONSE_EXPECT	Response expected flag. 0 : No response expected 1 : Response expected	1'b0
[5:0]	R/W	CMD_INDEX	Command index.	6'b0
SD/MMC Response Register 0 (RESP0)				
Address : SD0 : C006_2030h SD1: C006_8030h SD2 : C006_9030h				
[31:0]	R	RESPONSE0	Bit [31:0] of response	32'b0
SD/MMC Response Register 1 (RESP1)				
Address : SD0 : C006_2034h SD1: C006_8034h SD2 : C006_9034h				
[31:0]	R	RESPONSE1	Bit [63:32] of long response.	32'b0
SD/MMC Response Register 2 (RESP2)				
Address : SD0 : C006_2038h SD1: C006_8038h SD2 : C006_9038h				
[31:0]	R	RESPONSE2	Bit [95:64] of long response.	32'b0
SD/MMC Response Register 3 (RESP3)				
Address : SD0 : C006_203Ch SD1: C006_803Ch SD2 : C006_903Ch				
[31:0]	R	RESPONSE3	Bit [127:96] of long response.	32'b0
SD/MMC Masked Interrupt Status Register (MINTSTS)				
Address : SD0 : C006_2040h SD1: C006_8040h SD2 : C006_9040h				
[31:17]	-	RESERVED	Reserved	15'b0
[16]	R	SDIO_INTERRUPT	Interrupt from SDIO card; SDIO interrupt for card enabled only if corresponding sdio_int_mask bit is set in Interrupt mask register (mask bit 1 enables interrupt; 0 masks interrupt).	1'b0

Bit	R/W	Symbol	Description	Reset Value
			0 : No SDIO interrupt from card 1 : SDIO interrupt from card.	
[15]	R	EBEINT	End bit error(read), Write no CRC (write) interrupt.	1'b0
[14]	R	ACDINT	Auto command done interrupt.	1'b0
[13]	R	SBEINT	Start bit error interrupt(SBE), Busy Complete Interrupt (BCI)	1'b0
[12]	R	HLEINT	Hardware locked write error interrupt	1'b0
[11]	R	FRUNINT	FIFO underrun/overrun error interrupt	1'b0
[10]	R	HTOINT	Data starvation by host timeout interrupt (HTO), Volt_switch_int	1'b0
[9]	R	DRTOINT	Data read timeout interrupt	1'b0
[8]	R	RTOINT	Response timeout interrupt	1'b0
[7]	R	DCRCINT	Data CRC error interrupt	1'b0
[6]	R	RCRCINT	Response CRC error interrupt	1'b0
[5]	R	RXDRINT	Receive FIFO data request interrupt	1'b0
[4]	R	TXDRINT	Transmit FIFO data request interrupt	1'b0
[3]	R	DTOINT	Data transfer over interrupt	1'b0
[2]	R	CDINT	Command done interrupt	1'b0
[1]	R	REINT	Response error interrupt	1'b0
[0]	R	CDINT	Card detect	1'b0

SD/MMC Raw Interrupt Status Register (RINTSTS)

Address : SD0 : C006_2044h SD1: C006_8044h SD2 : C006_9044h

[31:17]	-	RESERVED	Reserved	15'b0
[16]	R/W	SDIO_INTERRUPT	SDIO interrupt from card	1'b0
[15]	R/W	EBE	End bit error(read), Write no CRC (write) interrupt.	1'b0
[14]	R/W	ACD	Auto command done interrupt.	1'b0
[13]	R/W	SBE	Start bit error interrupt(SBE), Busy Complete Interrupt (BCI)	1'b0
[12]	R/W	HLE	Hardware locked write error interrupt	1'b0
[11]	R/W	FRUN	FIFO underrun/overrun error interrupt	1'b0
[10]	R/W	HTO	Data starvation by host timeout interrupt (HTO), Volt_switch_int	1'b0
[9]	R/W	DRTO	Data read timeout interrupt	1'b0
[8]	R/W	RTO	Response timeout interrupt	1'b0
[7]	R/W	DCRC	Data CRC error interrupt	1'b0
[6]	R/W	RCRC	Response CRC error interrupt	1'b0
[5]	R/W	RXDR	Receive FIFO data request interrupt	1'b0
[4]	R/W	TXDR	Transmit FIFO data request interrupt	1'b0
[3]	R/W	DTO	Data transfer over interrupt	1'b0
[2]	R/W	CD	Command done interrupt	1'b0
[1]	R/W	RE	Response error interrupt	1'b0

Bit	R/W	Symbol	Description	Reset Value																
[0]	R	CDINT	Card detect	1'b0																
SD/MMC Status Register (STATUS)																				
Address : SD0 : C006_2048h SD1: C006_8048h SD2 : C006_9048h																				
[31]	R	DMA_REQ	DMA request signal state	1'b0																
[30]	R	DMA_ACK	DMA acknowledge signal state	1'b0																
[29:17]	R	FIFOCOUNT	Number of filled locations in FIFO	13'b0																
[16:11]	R	RESPONSE_INDEX	Index of previous response, including any auto-stop sent by core	6'b0																
[10]	R	DATA_STATE_MC_BUSY	Data transmit or receive state-machine is busy	1'b0																
[9]	R	DATA_BUSY	Selected card data busy 0 : card data not busy 1 : card data busy	1'bx																
[8]	R	DATA_3_STATUS	Checks whether card is present 0 : card not present 1 : card present	1'bx																
[7:4]	R	COMMAND FSM STATES	<p>Command FSM states.</p> <table> <tr><td>0 : Idle</td><td>1 : Send init sequence</td></tr> <tr><td>2 : Tx cmd start bit</td><td>3 : Tx cmd tx bit</td></tr> <tr><td>4 : Tx cmd index + arg</td><td>5 : Tx cmd crc7</td></tr> <tr><td>6 : Tx cmd end bit</td><td>7 : Rx resp start bit</td></tr> <tr><td>8 : Rx resp IRQ response</td><td>9 : Rx resp tx bit</td></tr> <tr><td>10 : Rx resp cmd idx</td><td>11 : Rx resp data</td></tr> <tr><td>12 : Rx resp crc7</td><td>13 : Rx resp end bit</td></tr> <tr><td>14 : Cmd path with NCC</td><td>15 : Wait</td></tr> </table> <p>NOTE: The command FSM state is represented using 19 bits.</p> <p>The STATUS Register(7:4) has 4 bits to represent the command FSM states. Using these 4 bits, only 16 states can be represented. Thus three states cannot be represented in the STATUS(7:4) register. The three states that are not represented in the STATUS Register(7:4) are:</p> <ul style="list-style-type: none"> * Bit 16 – Wait for CCS * Bit 17 – Send CCSD * Bit 18 – Boot Mode <p>Due to this, while command FSM is in "Wait for CCS state" or "Send CCSD" or "Boot Mode", the Status register indicates status as 0 for the bit field 7:4.</p>	0 : Idle	1 : Send init sequence	2 : Tx cmd start bit	3 : Tx cmd tx bit	4 : Tx cmd index + arg	5 : Tx cmd crc7	6 : Tx cmd end bit	7 : Rx resp start bit	8 : Rx resp IRQ response	9 : Rx resp tx bit	10 : Rx resp cmd idx	11 : Rx resp data	12 : Rx resp crc7	13 : Rx resp end bit	14 : Cmd path with NCC	15 : Wait	4'b0
0 : Idle	1 : Send init sequence																			
2 : Tx cmd start bit	3 : Tx cmd tx bit																			
4 : Tx cmd index + arg	5 : Tx cmd crc7																			
6 : Tx cmd end bit	7 : Rx resp start bit																			
8 : Rx resp IRQ response	9 : Rx resp tx bit																			
10 : Rx resp cmd idx	11 : Rx resp data																			
12 : Rx resp crc7	13 : Rx resp end bit																			
14 : Cmd path with NCC	15 : Wait																			
[3]	R	FIFO_FULL	FIFO is full	1'b0																
[2]	R	FIFO_EMPTY	FIFO is empty	1'b1																
[1]	R	FIFO_TX_WATERMARK	FIFO reached transmit watermark level. not qualified with data transfer.	1'b1																
[0]	R	FIFO_RX_WATERMARK	FIFO reached receive watermark level. not qualified with data transfer.	1'b0																
SD/MMC FIFO Threshold Watermark Register (FIFOTH)																				
Address : SD0 : C006_204Ch SD1: C006_804Ch SD2 : C006_904Ch																				
[31]	-	RESERVED	Reserved	1'b0																
[30:28]	R/W	DMA_MULTIPLE_TRANSACTION_SIZE	<p>Burst size of multiple transaction; should be programmed same as DW-DMA controller multiple-transaction-size SRC/DEST_MSIZE.</p> <table> <tr><td>000 : 1 transfers</td><td>001 : 4</td></tr> <tr><td>010 : 8</td><td>011 : 16</td></tr> <tr><td>100 : 32</td><td>101 : 64</td></tr> </table>	000 : 1 transfers	001 : 4	010 : 8	011 : 16	100 : 32	101 : 64	3'b0										
000 : 1 transfers	001 : 4																			
010 : 8	011 : 16																			
100 : 32	101 : 64																			

Bit	R/W	Symbol	Description	Reset Value
SD/MMC Card Detect Register (CDETECT) Address : SD0 : C006_2050h SD1: C006_8050h SD2 : C006_9050h				
[31:1]	-	RESERVED	Reserved	32'b0
[0]	R	CARD_DETECT_N	Value on card_detect_n input ports.	32'b0
SD/MMC Card Write Protect Register (WRTPRT) Address : SD0 : C006_2054h SD1: C006_8054h SD2 : C006_9054h				
[31:1]	-	RESERVED	Reserved	32'b0
[0]	R	WRITE_PROTECT	Value on card_write_prt input ports	32'b0
RESERVED Address : SD0 : C006_2058h SD1: C006_8058h SD2 : C006_9058h				
SD/MMC Transferred CIU Card Byte Count Register (TCBCNT) Address : SD0 : C006_205Ch SD1: C006_805Ch SD2 : C006_905Ch				
[31:0]	R	TRANS_CARD_BYTE_COUNT	Number of bytes transferred by CIU unit to card.	32'b0
SD/MMC Transferred Host to BIU-FIFO Byte Count Register (TBBCNT) Address : SD0 : C006_2060h SD1: C006_8060h SD2 : C006_9060h				
[31:0]	R	TRANS_FIFO_BYTE_COUNT	Number of bytes transferred between host/DMA memory and BIU FIFO	32'b0
SD/MMC Debounce Count Register (DEBNCE) Address : SD0 : C006_2064h SD1: C006_8064h SD2 : C006_9064h				
[31:24]	-	RESERVED	Reserved	8'b0
[23:0]	R/W	DEBOUNCE_COUNT	Number of host clocks (clk) used by debounce filter logic; typical debounce time is 5-25 ms.	24'hffff
SD/MMC User ID Register (USRID) Address : SD0 : C006_2068h SD1: C006_8068h SD2 : C006_9068h				
[31:0]	R/W	USRID	User identification register; value set by user	32'b0
SD/MMC Version ID Register (VERID) Address : SD0 : C006_206Ch SD1: C006_806Ch SD2 : C006_906Ch				
[31:0]	R/W	VERID	version identification register	32'h5342240a
RESERVED Address : SD0 : C006_2070h SD1: C006_8070h SD2 : C006_9070h				
SD/MMC UHS-1 Register (UHS_REG) Address : SD0 : C006_2074h SD1: C006_8074h SD2 : C006_9074h				
[31:17]	-	RESERVED	Reserved	15'b0
[16]	R/W	DDR_REG	DDR mode. Determines the voltage fed to the buffers by an external voltage regulator. 0 : Non-DDR mode 1 : DDR mode	1'b0
[15:1]	-	RESERVED	Reserved	15'b0
[0]	R/W	VOLT_REG	High Voltage mode. Determines the voltage fed to the buffers by an external voltage regulator. 0 : Buffers supplied with 3.3V Vdd 1 : Buffers supplied with 1.8V Vdd	1'b0

Bit	R/W	Symbol	Description	Reset Value
SD/MMC H/W Reset (RST_n)				
Address : SD0 : C006_2078h SD1: C006_8078h SD2 : C006_9078h				
[31:1]	-	RESERVED	Reserved	31'b0
[0]	R/W	CARD_RESET	Hardware reset. 1 : Active mode 0 : Reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized.	1'b0
SD/MMC Bus Mode Register (BMODE)				
Address : SD0 : C006_2080h SD1: C006_8080h SD2 : C006_9080h				
[31:1]	-	RESERVED	Reserved	21'b0
[10:8]	R/W	PBL	Programmable Burst Length. These bits indicate the maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of FIFO register. In order to change this value, write the required value to FIFO register. This is an encode value as follows. 000 – 1 transfers 001 – 4 transfers 010 – 8 transfers 011 – 16 transfers 100 – 32 transfers 101 – 64 transfers 110 – 128 transfers 111 – 256 transfers Transfer unit is either 16, 32, or 64 bits, based on HDATA_WIDTH. PBL is a read-only value and is applicable only for Data Access; it does not apply to descriptor accesses.	3'b0
[7]	R/W	DE	IDMAC Enable. When set, the IDMAC is enabled. DE is read/write.	1'b0
[6:2]	R/W	DSL	Descriptor Skip Length. Specifies the number of HWord/Word/Dword (depending on 16/32/64-bit bus) to skip between two unchained descriptors. This is applicable only for dual buffer structure. DSL is read/write.	5'b0
[1]	R/W	FB	Fixed Burst. Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations. FB is read/write.	1'b0
[0]	R/W	SWR	Software Reset. When set, the DMA Controller resets all its internal registers. SWR is read/write. It is automatically cleared after 1 clock cycle.	1'b0
SD/MMC Poll Demand Register (PLDMND)				
Address : SD0 : C006_2084h SD1: C006_8084h SD2 : C006_9084h				
[31:0]	W	PD	Poll Demand. If the OWN bit of a descriptor is not set, the FSM goes to the Suspend state. The host needs to write any value into this register for the IDMAC FSM to resume normal descriptor fetch operation. This is a write only register. PD bit is write-only.	32'b0
SD/MMC Descriptor List Base Address Register (DBADDR)				
Address : SD0 : C006_2088h SD1: C006_8088h SD2 : C006_9088h				
[31:0]	R/W	SDL	Start of Descriptor List. Contains the base address of the First Descriptor. The LSB bits [0/1/2:0] for 16/32/64-bit bus-width) are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only.	32'b0
SD/MMC Internal DMAC Status Register (IDSTS)				
Address : SD0 : C006_208Ch SD1: C006_808Ch SD2 : C006_908Ch				

Bit	R/W	Symbol	Description	Reset Value
[31:17]	-	RESERVED	Reserved	15'b0
[16:13]	R	FSM	<p>DMAC FSM present state.</p> <p>0 : DMA_IDLE 1 : DMA_SUSPEND 2 : DESC_RD 3 : DESC_CHK 4 : DMA_RD_REQ_WAIT 5 : DMA_WR_REQ_WAIT 6 : DMA_RD 7 : DMA_WR 8 : DESC_CLOSE</p> <p>This bit is read-only.</p>	4'b0
[12:10]	R	EB	<p>Error Bits. Indicates the type of error that caused a Bus Error. Valid only with Fatal Bus Error bit (IDSTS[2]) set. This field does not generate an interrupt.</p> <p>3'b001 : Host Abort received during transmission 3'b010 : Host Abort received during reception Others: Reserved</p> <p>EB is read-only.</p>	3'b0
[9]	R/W	AIS	<p>Abnormal Interrupt Summary. Logical OR of the following:</p> <ul style="list-style-type: none"> • IDSTS[2] – Fatal Bus Interrupt • IDSTS[4] – DU bit Interrupt <p>Only unmasked bits affect this bit.</p> <p>This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. Writing a 1 clears this bit.</p>	1'b0
[8]	R/W	NIS	<p>Normal Interrupt Summary. Logical OR of the following:</p> <ul style="list-style-type: none"> • IDSTS[0] – Transmit Interrupt • IDSTS[1] – Receive Interrupt <p>Only unmasked bits affect this bit.</p> <p>This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared. Writing a 1 clears this bit.</p>	1'b0
[7:6]	-	RESERVED	Reserved	2'b0
[5]	R/W	CES	<p>Card Error Summary. Indicates the status of the transaction to/from the card; also present in RINTSTS. Indicates the logical OR of the following bits:</p> <ul style="list-style-type: none"> • EBE – End Bit Error • RTO – Response Timeout/Boot Ack Timeout • RCRC – Response CRC • SBE – Start Bit Error • DRTO – Data Read Timeout/BDS timeout • DCRC – Data CRC for Receive • RE – Response Error <p>Writing a 1 clears this bit.</p> <p>The abort condition of the IDMAC depends on the setting of this CES bit. If the CES bit is enabled, then the IDMAC aborts on a “response error”; however, it will not abort if the CES bit is cleared.</p>	1'b0
[4]	R/W	DU	Descriptor Unavailable Interrupt. This bit is set when the descriptor is unavailable due to OWN bit = 0 (DESO[31]=0). Writing a 1 clears this bit.	1'b0
[3]	-	RESERVED	Reserved	1'b0
[2]	R/W	FBE	Fatal Bus Error Interrupt. Indicates that a Bus Error occurred (IDSTS[12:10]). When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit.	1'b0
[1]	R/W	RI	Receive Interrupt. Indicates the completion of data reception for a descriptor. Writing a 1 clears	1'b0

Bit	R/W	Symbol	Description	Reset Value
			this bit.	
[0]	R/W	TI	Transmit Interrupt. Indicates that data transmission is finished for a descriptor. Writing a '1' clears this bit.	1'b0
SD/MMC Internal DMAC Interrupt Enable Register (IDINTEN)				
Address : SD0 : C006_2090h SD1: C006_8090h SD2 : C006_9090h				
[31:10]	-	RESERVED	Reserved	22'b0
[9]	R/W	AI	Abnormal Interrupt Summary Enable. When set, an abnormal interrupt is enabled. This bit enables the following bits: • IDINTEN[2] : Fatal Bus Error Interrupt • IDINTEN[4] : DU Interrupt	1'b0
[8]	R/W	NI	Normal Interrupt Summary Enable. When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: • IDINTEN[0] : Transmit Interrupt • IDINTEN[1] : Receive Interrupt	1'b0
[7:6]	-	RESERVED	Reserved	2'b0
[5]	R/W	CES	Card Error summary Interrupt Enable. When set, it enables the Card Interrupt summary.	1'b0
[4]	R/W	DU	Descriptor Unavailable Interrupt. When set along with Abnormal Interrupt Summary Enable, the DU interrupt is enabled.	1'b0
[3]	-	RESERVED	Reserved	1'b0
[2]	R/W	FBE	Fatal Bus Error Enable. When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled.	1'b0
[1]	R/W	RI	Receive Interrupt Enable. When set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, Receive Interrupt is disabled.	1'b0
[0]	R/W	TI	Transmit Interrupt Enable. When set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled.	1'b0
SD/MMC Current Host Descriptor Address Register (DSCADDR)				
Address : SD0 : C006_2094h SD1: C006_8094h SD2 : C006_9094h				
[31:0]	R	HDA	Host Descriptor Address Pointer. Cleared on reset. Pointer updated by IDMAC during operation. This register points to the start address of the current descriptor read by the IDMAC.	32'b0
SD/MMC Current Buffer Descriptor Address Register (BUFADDR)				
Address : SD0 : C006_2098h SD1: C006_8098h SD2 : C006_9098h				
[31:0]	R	HBA	Host Buffer Address Pointer. Cleared on Reset. Pointer updated by IDMAC during operation. This register points to the current Data Buffer Address being accessed by the IDMAC.	32'b0
RESERVED				
Address : SD0 : C006_209Ch SD1: C006_809Ch SD2 : C006_909Ch				
SD/MMC Card Threshold Control Register (CARDTHRCTL)				
Address : SD0 : C006_2100h SD1: C006_8100h SD2 : C006_9100h				
[31:27]	-	RESERVED	Reserved	5'b0
[26:16]	R/W	CARD_RD_THRESHOLD	Card Read Threshold size; N depends on the FIFO size.	11'b0
[15:2]	-	RESERVED	Reserved	14'b0
[1]	R/W	BSY_CLR_INTEN	Busy Clear Interrupt generation: • 0 : Busy Clear Interrupt disabled • 1 : Busy Clear Interrupt enabled	1'b0

Bit	R/W	Symbol	Description	Reset Value
			Note: The application can disable this feature if it does not want to wait for a Busy Clear Interrupt. For example, in a multi-card scenario, the application can switch to the other card without waiting for a busy to be completed. In such cases, the application can use the polling method to determine the status of busy. By default this feature is disabled and backward-compatible to the legacy drivers where polling is used.	
[0]	R/W	CARD_RD_THREN	Card Read Threshold Enable <ul style="list-style-type: none"> • 1'b0 - Card Read Threshold disabled • 1'b1 - Card Read Threshold enabled. Host Controller initiates Read Transfer only if CardRdThreshold amount of space is available in receive FIFO. 	1'b0
SD/MMC Back-end Power Register (BACK_END_POWER)				
Address : SD0 : C006_2104h SD1: C006_8104h SD2 : C006_9104h				
[31:1]	-	RESERVED	Reserved	31'b0
[0]	R/W	BACK_END_POWER	Back end power <ul style="list-style-type: none"> • 1'b0 : Off; Reset • 1'b1 : Back-end Power supplied to card application; one pin per card 	1'b0
RESERVED				
Address : SD0 : C006_2108h SD1: C006_8108h SD2 : C006_9108h				
SD/MMC eMMC 4.5 DDR START Bit Detection Control Register (EMMC_DDR_REG)				
Address : SD0 : C006_210Ch SD1: C006_810Ch SD2 : C006_910Ch				
[31:1]	-	RESERVED	Reserved	31'b0
[0]	R/W	HALF_START_BIT	Control for start bit detection mechanism inside DWC_mobile_storage based on duration of start bit; each bit refers to one slot. For eMMC 4.5, start bit can be: <ul style="list-style-type: none"> • Full cycle (HALF_START_BIT = 0) • Less than one full cycle (HALF_START_BIT = 1) Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications.	1'b0
RESERVED				
Address : SD0 : C006_2110h SD1: C006_8110h SD2 : C006_9110h				
SD/MMC Enable Phase Shift Register (EMMC_DDR_REG)				
Address : SD0 : C006_2114h SD1: C006_8114h SD2 : C006_9114h				
[31:26]	-	RESERVED	Reserved	6'b0
[25:24]	R/W	PHASE_SHIFT_SAMPLE	Sample clock phase shift 0:0 1:90 2:180 3:270	2'b0
[23:18]	-	RESERVED	Reserved	6'b0
[17:16]	R/W	PHASE_SHIFT_DRIVE	Drive clock phase shift 0:0 1:90 2:180 3:270	2'b0
[15:8]	R/W	DELAY_SAMPLE	Sample clock delay	8'b0
[7:0]	R/W	DELAY_DRIVE	Drive clock delay	8'b0
RESERVED				
Address : SD0 : C006_2118h~C006_21FF SD1: C006_8110h~C006_81FFh SD2 : C006_9110h~ C006_9110h				

Bit	R/W	Symbol	Description	Reset Value
SD/MMC Data Register				
Address : SD0 : C006_2200h SD1: C006_8200h SD2 : C006_9200h				
[31:0]	R/W	DATA	Data write to or read from FIFO	32'bx

Section 18. Pulse Period Measurement (PPM)

18.1 Overview

The Pulse Period Measurement (PPM) measures the period of the high level and the low level of a 1-bit Signal entered from the outside.

18.1.1 Features

The PPM provides:

- 16-bit Pulse Period Measurement Counter
- Overflow Check
- Control Input Polarity
- PPM Clock generator

18.1.2 Block Diagram

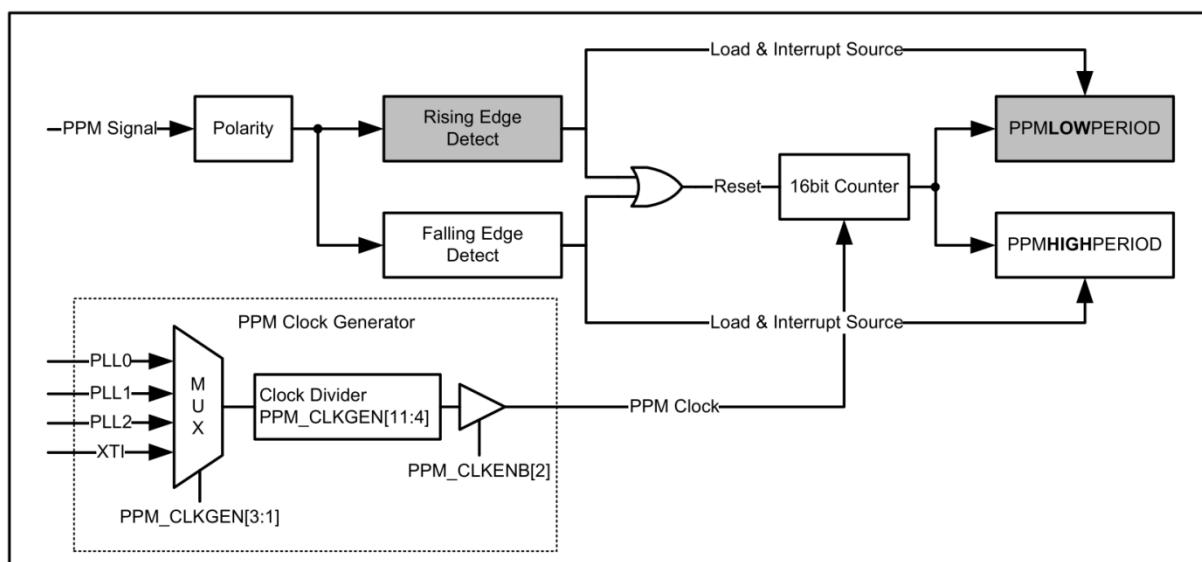


Figure 18-1. PPM Block Diagram

- **PPMLOWPERIOD** : PPM Low Period Register : If PPM Signal is changed from Low to High (rising edge detect), the count value of 16-bit Counter is stored at PPMLOWPERIOD Register and 16-bit Counter is reset.
- **PPMHIGHPERIOD** : PPM High Period Register : If PPM Signal is changed from High to Low (falling edge detect), the count value of 16-bit Counter is stored at PPMHIGHPERIOD Register and 16-bit Counter is reset.
- If the high or low period is too long to be measured by the 16-bit Counter, Overflow interrupt occurs.

18.2 Functional Description

When a PPM signal rising edge or a falling edge is detected, the 16-bit counter is reset after storing the 16-bit counter value to the PPMLOWPERIOD register or the PPMHIGHPERIOD register. If a falling edge is detected, the counter value is stored at the PPMHIGHPERIOD register and then the counter is reset. If a rising edge is detected, the counter value is stored at the PPMLOWPERIOD register and then the counter is reset. If the high or the low period is too long to be measured by the 16-bit Counter, an Overflow interrupt occurs. Therefore the IP-Remocon input signal should be checked and an appreciate PPL clock value should be specified to prevent the occurrence of Overflow interrupts.

18.2.1 IR Remote Protocol Example

Figure 18-2 shows a representative waveform of an IP Remote signal being passed IR-Receiver Module.

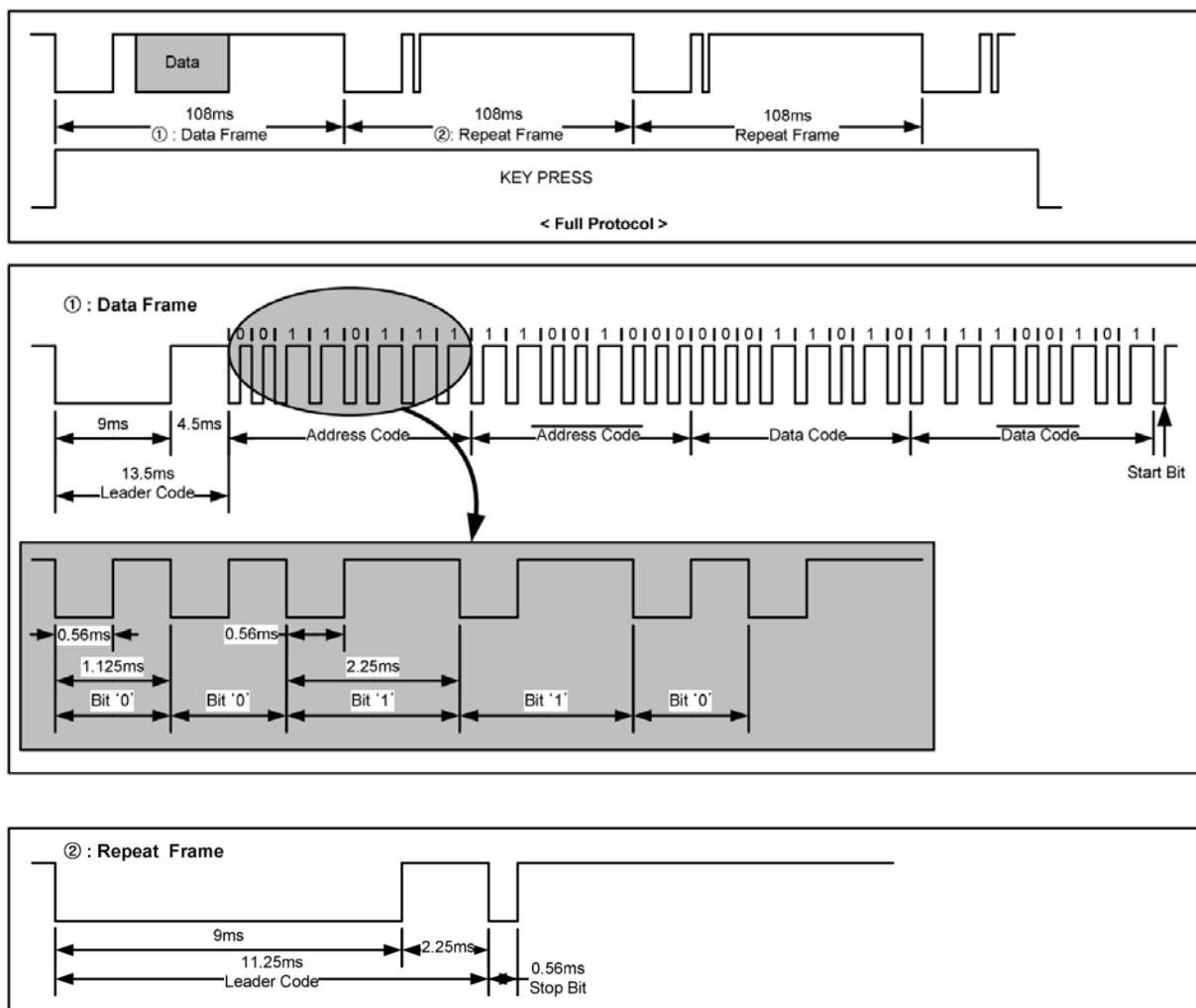


Figure 18-2. IR Remote Example Protocol

The PPM clock between 850 KHz and 6.75 MHz is suitable to prevent the occurrence of an Overflow interrupt.

The PPM clock can be selected in the range of 843,750 Hz to 13,500,000 Hz via the Clock Divider setting.(1~31)

18.2.2 Timing

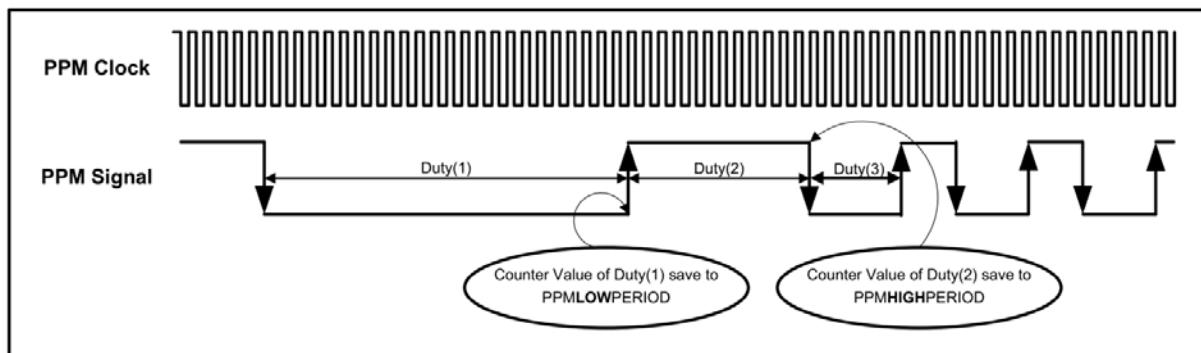


Figure 18-3. PPM Timing

18.2.3 Flowchart

Figure 18-4 shows an example of a flowchart for checking IR-Remote signals.

Initialization Procedure for PPM :

1. Set GPIO
2. Clock Source Select (XTI)
3. Clock Divider (1~31)
4. Set Polarity
5. Set Interrupt mode (Falling Edge, Rising Edge and Overflow)
6. Initialize the PPM Status Register.
7. Enable Interrupt

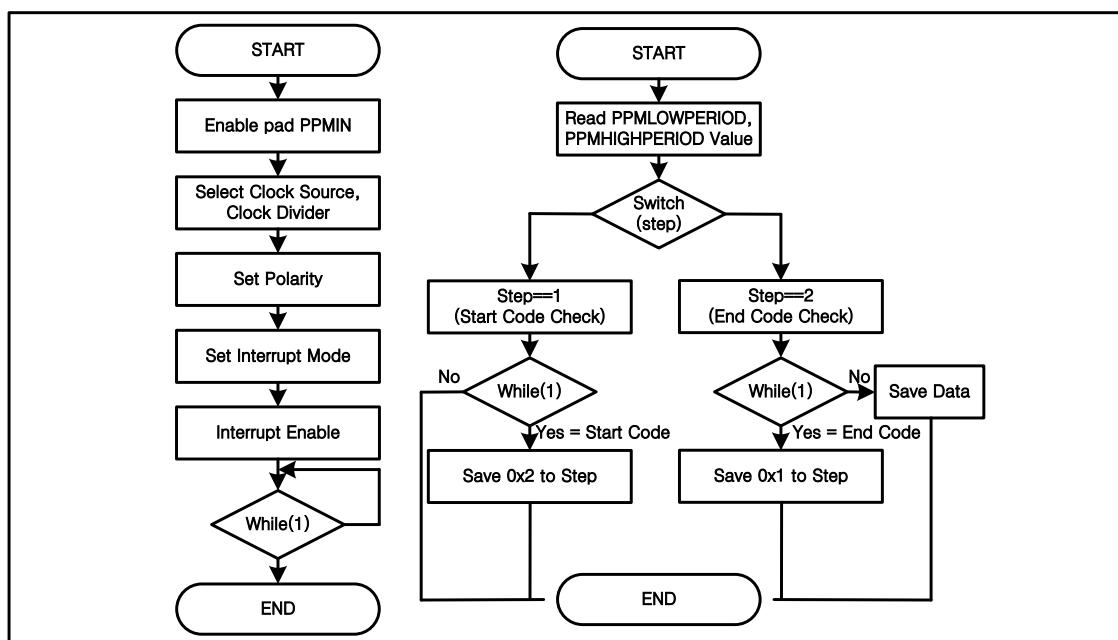


Figure 18-4. IR Remote Receiver Flowchart

18.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
PPM CONTROL REGISTER (PPMCTRL)				
<i>Address : C0054000h</i>				
[31:16]	-	RESERVED	Reserved	-
[15]	R/W	PPMENB	PPM Module Enable 0: Disable 1: Enable	1'b0
[14]	R/W	PPMINPOL	Control Polarity of PPM Input Signal 0: Invert 1: Bypass	1'b1
[31:16]	-	RESERVED	Reserved	-
[2]	R/W	PPMIRQO	Overflow Interrupt Enable 0: Disable 1: Enable	1'b0
[1]	R/W	PPMIRQF	Falling Edge Detect Interrupt Enable 0: Disable 1: Enable	1'b0
[0]	R/W	PPMIRQR	Rising Edge Detect Interrupt Enable 0: Disable 1: Enable	1'b0
PPM STATUS REGISTER (PPMSTATUS)				
<i>Address : C0054004h</i>				
[31:5]	-	RESERVED	Reserved	-
[4]	R	PPMOVERLOW	Indicate Overflow in Low 0: Normal 1: Overflow	1'b0
[3]	R	PPMOVERHIGH	Indicate Overflow in High 0: Normal 1: Overflow	1'b0
[2]	R/W	PPMPENDO	Overflow Interrupt Pending Bit. Set to 1 to clear this bit 0: None 1: Detected	1'b0
[1]	R/W	PPMPENDF	Falling Edge Detect Interrupt Pending Bit. Set to 1 to clear this bit 0: None 1: Detected	1'b0
[0]	R/W	PPMPENDR	Rising Edge Detect Interrupt Pending Bit. Set to 1 to clear this bit 0: None 1: Detected	1'b0
PPM LOW PERIOD REGISTER (PPMLOWPERIOD)				
<i>Address : C0054006h</i>				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R	PPMLOWPERIOD	Read Counter Value of Low	16'h0
PPM HIGH PERIOD REGISTER (PPMHIGHPERIOD)				
<i>Address : C0054008h</i>				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R	PPMHIGHPERIOD	Read Counter Value of High	16'h0

Section 19. *Pulse Width Modulation (PWM) Timer*

19.1 Overview

These timers can be used to generate internal interrupts to the ARM subsystem. In addition, Timers 0, 1, 2, and 3 include a PWM function (Pulse Width Modulation) which can drive an external I/O signal. The PWM for timer 0 has a optional dead-zone generator capability, which can be utilized to support a large current device. Timer 4 is only an internal timer with no output pins.

The Timers are normally clocked off of a divided version of the APB-PCLK. Timers 0 and 1 share a programmable 8-bit prescaler that provides the first level of division for the PCLK. Timer 2, 3, and 4 share a different 8-bit prescaler. Each timer has its own, private clock-divider that provides a second level of clock division (prescaler divided by 2,4,8, or 16). Alternatively, the Timers can select a clock source from an external pin. Timers 0 and 1 can select the external clock TCLK0. Timers 2, 3, and 4 can select the external clock TCLK1.

Each timer has its own 32-bit down-counter which is driven by the timer clock. The down-counter is initially loaded from the Timer Count Buffer register (TCNTBn). When the down-counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation is completed. When the timer down-counter reaches zero, the value of corresponding TCNTBn can be automatically reloaded into the down-counter to start the next cycle. However, if the timer stops, for example, by clearing the timer enable bit of TCONn during the timer running mode, the value of TCNTBn will not be reloaded into the counter.

The Pulse Width Modulation function (PWM) uses the value of the TCMPBn register. The timer control logic changes the output level when the down-counter value matches the value of the compare register in the timer control logic. Therefore, the compare register determines the turn-on time(or turn-off time) of a PWM output.

The TCNTBn and TCMPBn registers are double buffered to allow the timer parameters to be updated in the middle of a cycle. The new values will not take effect until the current timer cycle completes.

A simple example of a PWM cycle is shown in the figure below.

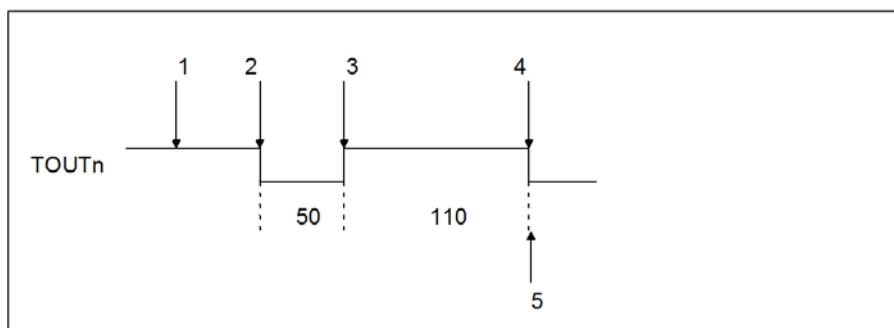


Figure 19-1. Simple Example of PWM Cycle

1. Initialize the TCNTBn with 160(50+110) and the TCMPBn with 110.
2. Start Timer by setting the start bit and manualupdate bit off. The TCNTBn value of 160 is loaded into the downcounter, the output is driven low.
3. When downcounter counts down to the value in the TCMPBn register (110), the output is changed from low to high
4. When the downcounter reaches 0, the interrupt request is generated.
5. At the same time the downcounter is automatically reloaded with TCNTBn, which restarts the cycle.

19.1.1 Features

- The Features supported by the PWM are:
 - Five 32-bit Timers.
 - Two 8-bit Clock Prescalers providing first level of division for the PCLK, Five Clock Dividers and Multiplexers providing second level of division for the Prescaler clock and Two External Clocks.
 - Programmable Clock Select Logic for individual PWM Channels.
 - Four Independent PWM Channels with Programmable Duty Control and Polarity.
 - Static Configuration: PWM is stopped.
 - Dynamic Configuration: PWM is running.
 - Supports Auto-Reload Mode and One-Shot Pulse Mode.
 - Supports for two external inputs to start PWM.
 - Dead Zone Generator on two PWM Outputs.
 - Supports DMA Transfers.
 - Optional Pulse or Level Interrupt Generation.
- The PWM has two operation modes:
 - Auto-Reload Mode:
 - Continuous PWM pulses are generated based on programmed duty cycle and polarity.
 - One-Shot Pulse Mode:
 - Only one PWM pulse is generated based on programmed duty cycle and polarity.

To control the functionality of PWM, 18 special function registers are provided. The PWM is a programmable output, dual clock input AMBA slave module and connects to the Advanced Peripheral Bus (APB). The 18 special function registers within PWM are accessed via APB transactions.

19.1.2 Block Diagram

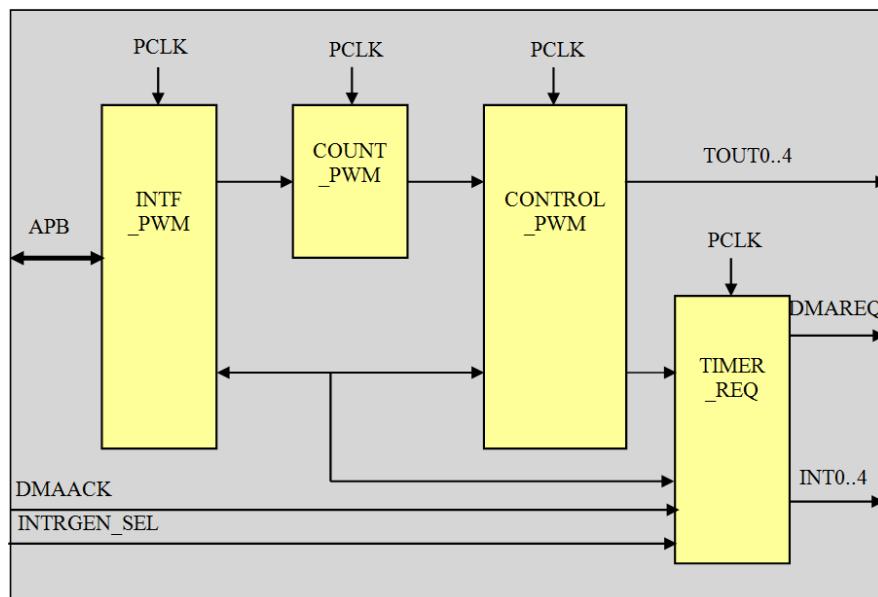


Figure 19-2. PWMTIMER Block Diagram

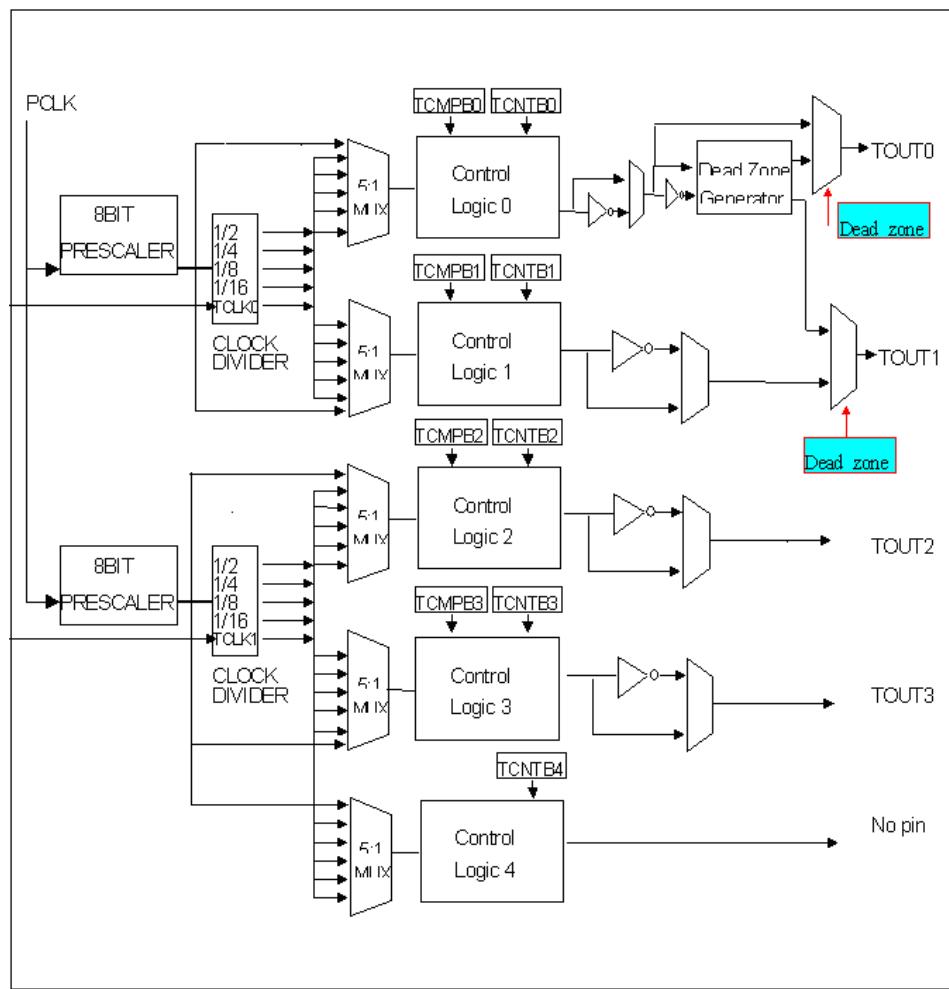


Figure 19-3. PWMTIMER Clock Tree Diagram

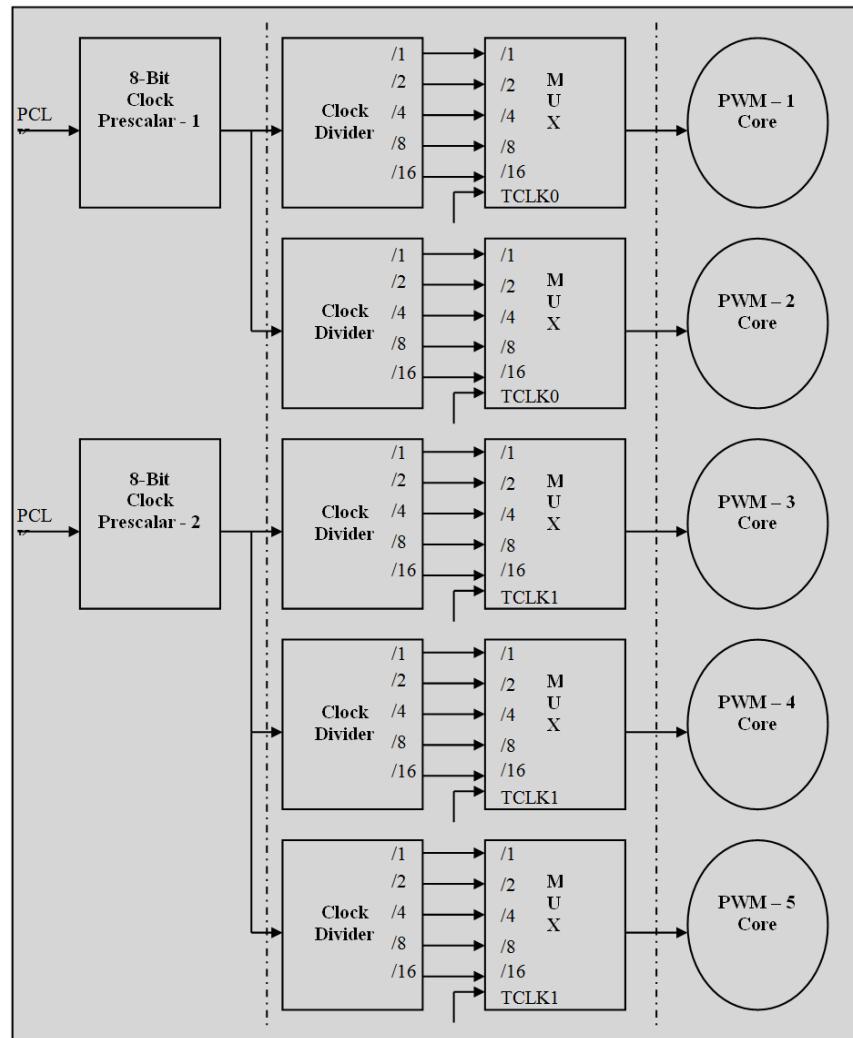


Figure 19-4. PWMTIMER Detailed Clock Tree Diagram

19.2 Functional Description

19.2.1 PRESCALER & DIVIDER

8-bit prescaler and 3-bit divider make the following output frequencies:

4-bit divider settings	minimum resolution (prescaler=0)	maximum resolution (prescaler=255)	maximum interval (TCNTBn=65535)
1/1(PCLK=66Mhz)	0.015us(66.0Mhz)	3.87us(258Khz)	0.25s
1/2 (PCLK=66Mhz)	0.030us(33.0Mhz)	7.75us(129Khz)	0.50s
1/4 (PCLK=66Mhz)	0.060us (16.5Mhz)	15.5us(64.5Khz)	1.02s
1/8 (PCLK=66Mhz)	0.121us (8.25Mhz)	31.0us (32.2Khz)	2.03s
1/16 (PCLK=66Mhz)	0.242us (4.13Mhz)	62.1us (16.1Khz)	4.07s

Table 19-1. Min. and Max. Resolution based on Prescaler and Clock Divider Values

19.2.2 BASIC TIMER OPERATION

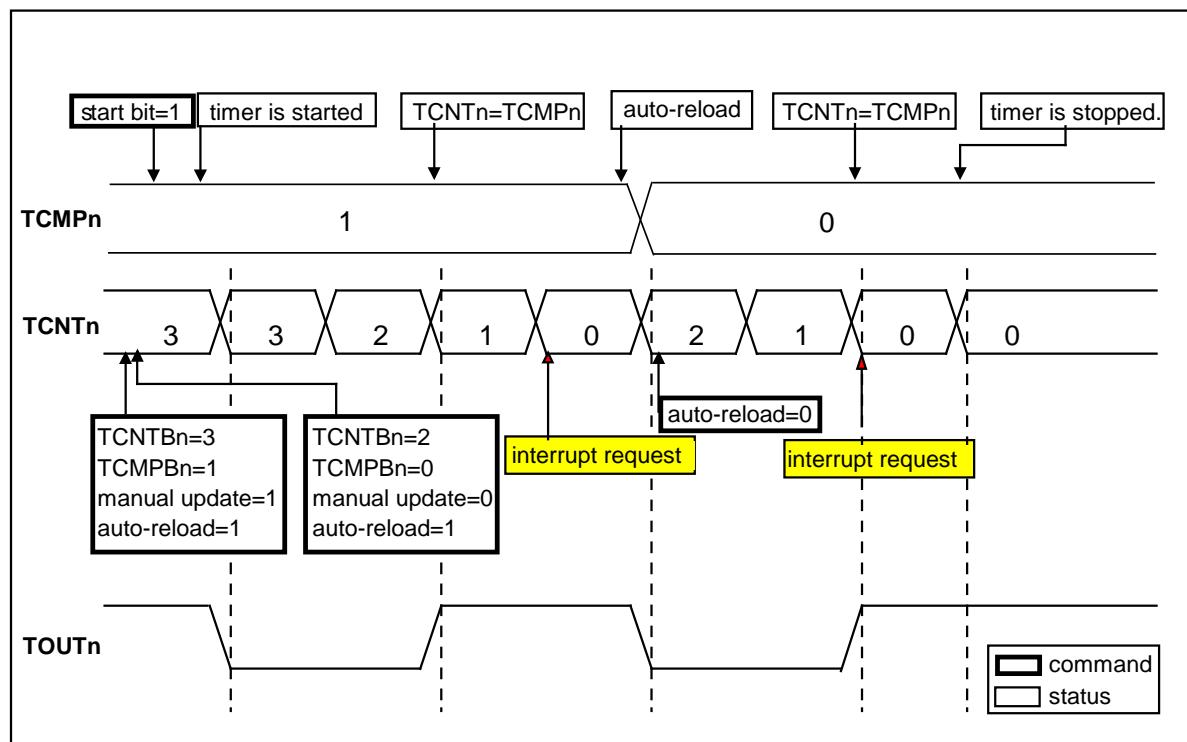


Figure 19-5. Timer operations

A timer (except the timer channel 5) has TCNTBn, TCNTn, TCMPBn and TCMPn. TCNTBn and TCMPBn are loaded into TCNTn and TCMPn when the timer reaches 0. When TCNTn reaches 0, the interrupt request will occur if the interrupt is enabled. (TCNTn and TCMPn are the names of the internal registers. The TCNTn register can be read from the TCNTOn register)

19.2.3 AUTO-RELOAD AND DOUBLE BUFFERING

The Timers have a double buffering feature, which can change the reload value for the next timer operation without stopping the current timer operation. So, although the new timer value is set, a current timer operation is completed successfully.

The timer value can be written into TCNTBn (Timer Count Buffer register) and the current counter value of the timer can be read from TCNTOn (Timer Count Observation register). If TCNTBn is read, the read value is not the current state of the counter but the reload value for the next timer duration.

The auto-reload is the operation, which copies the TCNTBn into TCNTn when TCNTn reaches 0. The value, written into TCNTBn, is loaded to TCNTn only when the TCNTn reaches to 0 and auto-reload is enabled. If the TCNTn is 0 and the auto-reload bit is 0, the TCNTn does not operate any further.

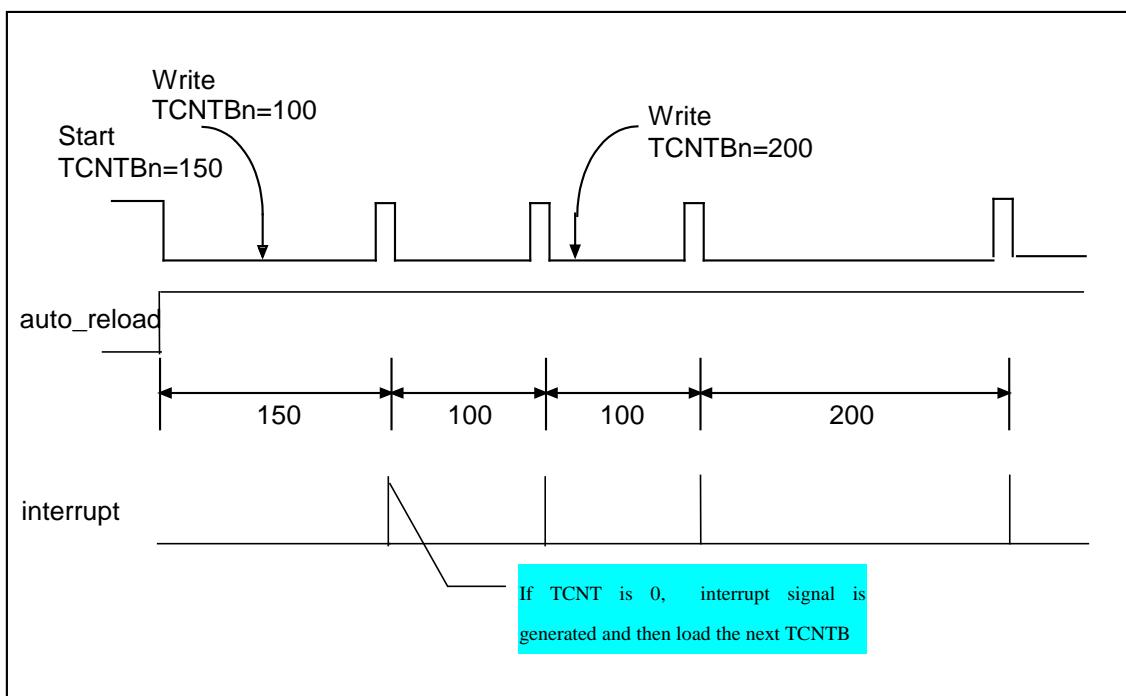


Figure 19-6. Example of Double Buffering Feature

The result of the following procedure is shown in Figure 19-7.

1. Enable the auto-reload feature. Set the TCNTBn as 160(50+110) and the TCMPBn as 110. Set the manual update bit and inverter bit(on/off). The manual update bit makes the TCNTn,TCMPn set to the value of TCNTBn,TCMPBn. And then, set TCNTBn,TCMPBn as 80(40+40),40 to determine the next reload value.
2. Start Timer by setting the start bit and manualupdata bit off.
3. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
4. When TCNTn reaches to 0, TCNTn is reloaded automatically with TCNTBn. At the same time, the interrupt request is generated.
5. In the ISR(interrupt service routine), the TCNTBn and TCMPBn is set as 80(20+60) and 60,which is used for next duration.
6. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
7. When TCNTn reaches to 0, TCNTn is reloaded automatically with TCNTBn. At the same time, the interrupt request is

generated.

8. In the ISR(interrupt service routine), auto-reload and interrupt request are disabled to stop the timer.
9. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
10. Even when TCNTn reaches to 0, TCNTn is not any more reloaded and the timer is stopped because auto-reload is disabled.
11. No interrupt request is generated.

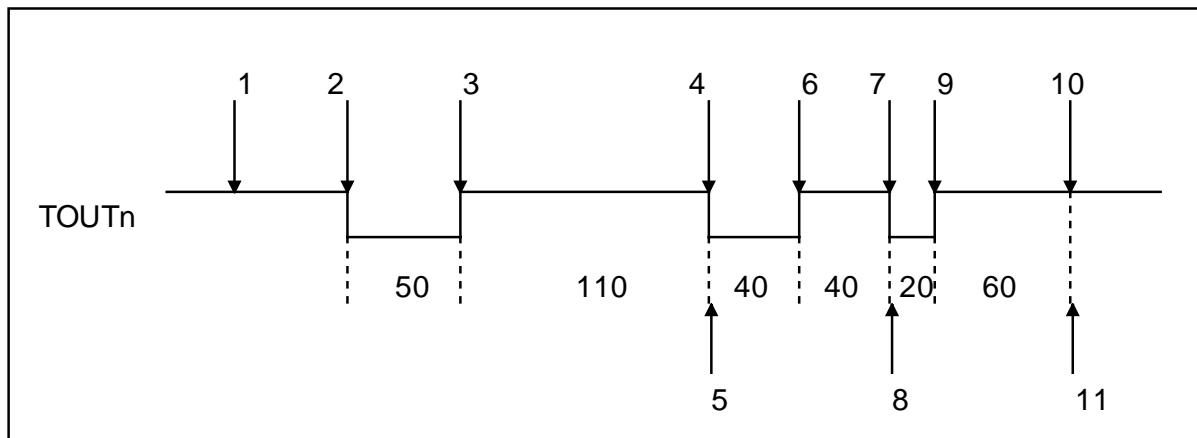


Figure 19-7. Example of Timer Operation

19.2.4 INITIALIZE TIMER (SETTING MANUAL-UP DATA AND INVERTER)

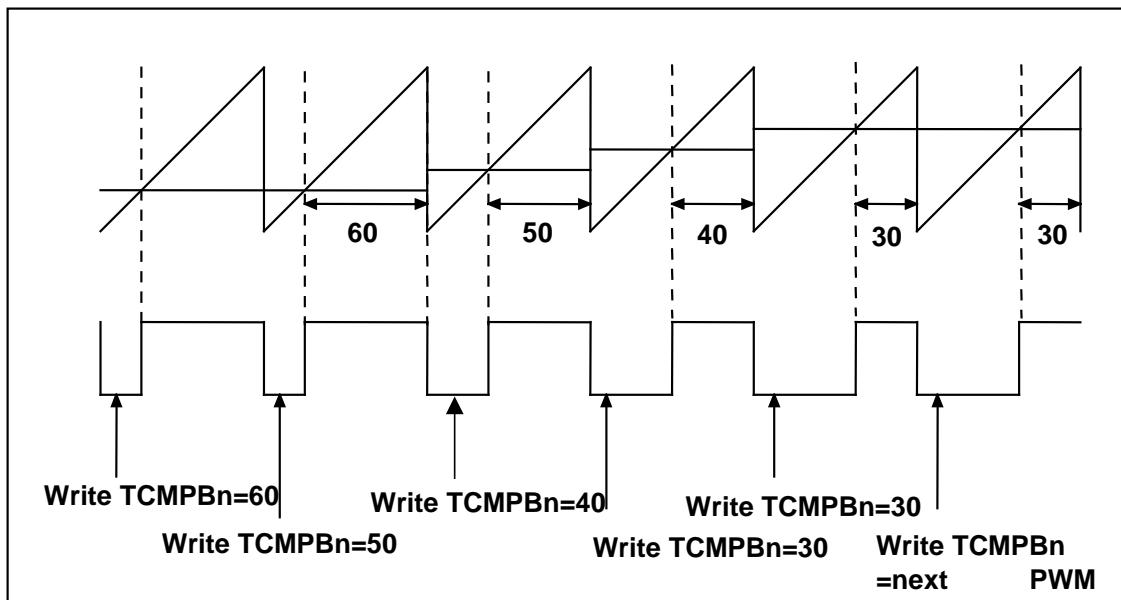


Figure 19-8. Example of PWM

PWM feature can implement by using the TCMPBn. PWM frequency is determined by TCNTBn. A PWM value is determined by TCMPBn in the Figure 19-8.

For higher PWM value, decrease TCMPBn value. For lower PWM value, increase TCMPBn value. If output inverter is enabled, the increment/decrement may be opposite.

Because of double buffering feature, TCMPBn, for a next PWM cycle, can be written in any point of current PWM cycle by ISR.

19.2.5 OUTPUT LEVEL CONTROL

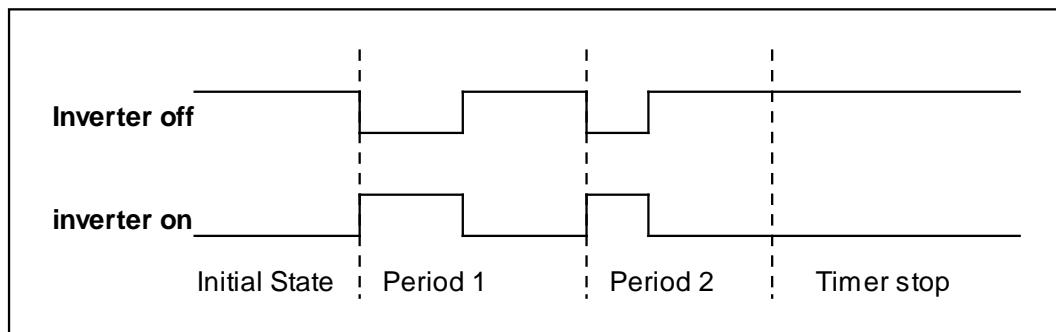


Figure 19-9. Inverter on/off

The following methods can be used to maintain TOUT as high or low.(assume the inverter is off)

1. Turn off the auto-reload bit. And then, TOUTn goes to high level and the timer is stopped after TCNTn reaches to 0. This method is recommended.
2. Stop the timer by clearing the timer start/stop bit to 0. If TCNTn <= TCMPn, the ouput level is high. If TCNTn >TCMPn, the output level is low
3. TOUTn can be inverted by the inverter on/off bit in TCON. The inverter removes the additional circuit to adjust the output level.

19.2.6 DEAD ZONE GENERATOR

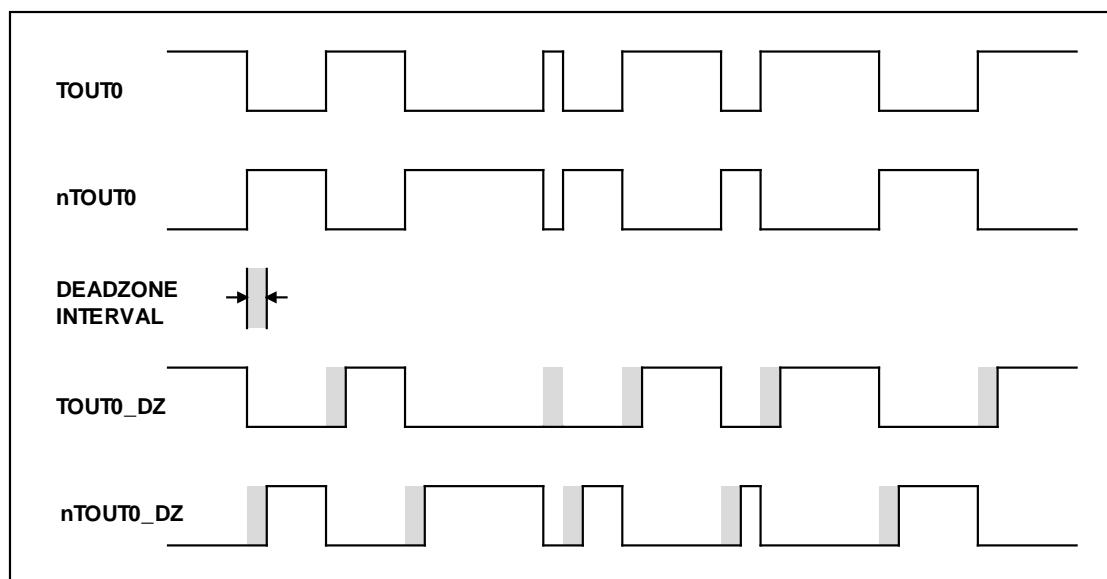


Figure 19-10. The waveform when a deadzone feature is enabled.

The deadzone is for the PWM control of power devices. This feature is used to insert the time gap between a turn-off of

a switching device and a turn on of the other switching device. This time gap prohibit the two switching device turning on simultaneously even for a very short time.

TOUT0 is the PWM output. nTOUT0 is the inversion of the TOUT0. If the dead-zone is enabled, the output wave-form of TOUT0,nTOUT0 will be TOUT0_DZ and nTOUT0_DZ. TOUT0_DZ and nTOUT0_DZ never can be turned on simultaneously by the dead zone interval. For functional correctness, the deadzonelength must be set smaller than compare counter value.

19.2.7 TIMER INTERRUPT GENERATION

The PWMTIMER provides flexibility to generate Pulse and Level Interrupts by controlling the 'INTRGEN_SEL' port status. When the port 'INTRGEN_SEL' is tied to logic 1, optional level interrupts will be generated else optional pulse interrupts will be generated. The interrupt generation is controlled by writing specific values to the 'TINT_CSTAT' register within PWMTIMER. Also, interrupt generations is optional based on programmed value in 'TINT_CSTAT' register.

19.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
Clock-Prescalar and Dead-Zone Configurations (TCFG0)				
<i>Address : TIMER : C0017000h / PWM : C0018000h</i>				
[31:24]	-	RESERVED	Reserved	-
[23:16]	R/W	DEAD ZONE LENGTH	Dead zone length	8'b0
[15:8]	R/W	PRESCALER 1	Prescaler 1 value for Timer 2, 3 and 4	8'b1
[7:0]	R/W	PRESCALER 0	Prescaler 0 value for timer 0 & 1	8'b1
Clock Multiplexers and DMA Mode Select (TCFG1)				
<i>Address : TIMER : C0017004h / PWM : C0018004h</i>				
[31:24]	-	RESERVED	Reserved	-
[23:20]	R/W	DMA MODE	Select DMA Request Channel Select Bit 0000: No select 0001: INT0 0010: INT1 0011: INT2 0100: INT3 0101: INT4 0110: No select 0111: No select	4'b0
[19:16]	R/W	DIVIDER MUX4	Select Mux input for PWM Timer 4 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	4'b0
[15:12]	R/W	DIVIDER MUX3	Select Mux input for PWM Timer 3 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	4'b0
[11:8]	R/W	DIVIDER MUX2	Select Mux input for PWM Timer 2 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	4'b0
[7:4]	R/W	DIVIDER MUX1	Select Mux input for PWM Timer 1 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	4'b0
[3:0]	R/W	DIVIDER MUX0	Select Mux input for PWM Timer 0 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	4'b0

Bit	R/W	Symbol	Description	Reset Value
Timer Control Register (TCON)				
<i>Address : TIMER : C0017008h / PWM : C0018008h</i>				
[31:23]	-	RESERVED	Reserved	-
[22]	R/W	TIMER 4 AUTO RELOAD ON/OFF	0: One-Shot 1: Interval Mode(Auto-Reload)	1'b0
[21]	R/W	TIMER 4 MANUAL UPDATE	0: No Operation 1: Update TCNTB4	1'b0
[20]	R/W	TIMER 4 START/STOP	0: Stop 1: Start Timer 4	1'b0
[19]	R/W	TIMER 3 AUTO RELOAD ON/OFF	0: One-Shot 1: Interval Mode(Auto-Reload)	1'b0
[18]	R/W	TIMER 3 OUTPUT INVERTER ON/OFF	0: Inverter Off 1: TOUT3 Inverter-On	1'b0
[17]	R/W	TIMER 3 MANUAL UPDATE	0: No Operation 1: Update TCNTB3,TCMPB3	1'b0
[16]	R/W	TIMER 3 START/STOP	0: Stop 1: Start Timer 3	1'b0
[15]	R/W	TIMER 2 AUTO RELOAD ON/OFF	0: One-Shot 1: Interval Mode(Auto-Reload)	1'b0
[14]	R/W	TIMER 2 OUTPUT INVERTER ON/OFF	0: Inverter Off 1: TOUT2 Inverter-On	1'b0
[13]	R/W	TIMER 2 MANUAL UPDATE	0: No Operation 1: Update TCNTB2,TCMPB2	1'b0
[12]	R/W	TIMER 2 START/STOP	0: Stop 1: Start Timer 2	1'b0
[11]	R/W	TIMER 1 AUTO RELOAD ON/OFF	0: One-Shot 1: Interval Mode(Auto-Reload)	1'b0
[10]	R/W	TIMER 1 OUTPUT INVERTER ON/OFF	0: Inverter Off 1: TOUT1 Inverter-On	1'b0
[9]	R/W	TIMER 1 MANUAL UPDATE	0: No Operation 1: Update TCNTB1,TCMPB1	1'b0
[8]	R/W	TIMER 1 START/STOP	0: Stop 1: Start Timer 1	1'b0
[7:5]	-	RESERVED	Reserved	-
[4]	R/W	DEAD ZONE ENABLE/DISABLE	Deadzone Generator Enable/Disable	1'b0
[3]	R/W	TIMER 0 AUTO RELOAD ON/OFF	0: One-Shot 1: Interval Mode(Auto-Reload)	1'b0
[2]	R/W	TIMER 0 OUTPUT INVERTER ON/OFF	0: Inverter Off 1: TOUT0 Inverter-On	1'b0
[1]	R/W	TIMER 0 MANUAL UPDATE	0: No Operation 1: Update TCNTB0,TCMPB0	1'b0
[0]	R/W	TIMER 0 START/STOP	0: Stop 1: Start Timer 0	1'b0
Timer 0 Count Buffer Register (TCNTB0)				
<i>Address : TIMER : C001700Ch / PWM : C001800Ch</i>				
[31:0]	R/W	TIMER 0 COUNT BUFFER	Timer 0 Count Buffer Register	32b0
Timer 0 Compare Buffer Register (TCMPB0)				
<i>Address : TIMER : C0017010h / PWM : C0018010h</i>				
[31:0]	R/W	TIMER 0 COMPARE BUFFER	Timer 0 Compare Buffer Register	32b0
Timer 0 Count Observation Register (TCNTO0)				
<i>Address : TIMER : C0017014h / PWM : C0018014h</i>				

Bit	R/W	Symbol	Description	Reset Value
[31 : 0]	R	TIMER 0 COUNT OBSERVATION	Timer 0 Count Observation Register	32b0
Timer 1 Count Buffer Register (TCNTB1)				
<i>Address : TIMER : C0017018h / PWM : C0018018h</i>				
[31:0]	R/W	TIMER 1 COUNT BUFFER	Timer 1 Count Buffer Register	32b0
Timer 1 Compare Buffer Register (TCMPB1)				
<i>Address : TIMER : C001701Ch / PWM : C001801Ch</i>				
[31 : 0]	R/W	TIMER 1 COMPARE BUFFER	Timer 1 Compare Buffer Register	32b0
Timer 1 Count Observation Register (TCNTO1)				
<i>Address : TIMER : C0017020h / PWM : C0018020h</i>				
[31 : 0]	R	TIMER 1 COUNT OBSERVATION	Timer 1 Count Observation Register	32b0
Timer 2 Count Buffer Register (TCNTB2)				
<i>Address : TIMER : C0017024h / PWM : C0018024h</i>				
[31:0]	R/W	TIMER 2 COUNT BUFFER	Timer 2 Count Buffer Register	32b0
Timer 2 Compare Buffer Register (TCMPB2)				
<i>Address : TIMER : C0017028h / PWM : C0018028h</i>				
[31 : 0]	R/W	TIMER 2 COMPARE BUFFER	Timer 2 Compare Buffer Register	32b0
Timer 2 Count Observation Register (TCNTO2)				
<i>Address : TIMER : C001702Ch / PWM : C001802Ch</i>				
[31 : 0]	R	TIMER 2 COUNT OBSERVATION	Timer 2 Count Observation Register	32b0
Timer 3 Count Buffer Register (TCNTB3)				
<i>Address : TIMER : C0017030h / PWM : C0018030h</i>				
[31:0]	R/W	TIMER 3 COUNT BUFFER	Timer 3 Count Buffer Register	32b0
Timer 3 Compare Buffer Register (TCMPB3)				
<i>Address : TIMER : C0017034h / PWM : C0018034h</i>				
[31 : 0]	R/W	TIMER 3 COMPARE BUFFER	Timer 3 Compare Buffer Register	32b0
Timer 3 Count Observation Register (TCNTO3)				
<i>Address : TIMER : C0017038h / PWM : C0018038h</i>				
[31 : 0]	R	TIMER 3 COUNT OBSERVATION	Timer 3 Count Observation Register	32b0
Timer 4 Count Buffer Register (TCNTB4)				
<i>Address : TIMER : C001703Ch / PWM : C001803Ch</i>				
[31:0]	R/W	TIMER 4 COUNT BUFFER	Timer 4 Count Buffer Register	32b0
Timer 4 Count Observation Register (TCNTO4)				
<i>Address : TIMER : C0017040h / PWM : C0018040h</i>				
[31 : 0]	R	TIMER 4 COUNT OBSERVATION	Timer 4 Count Observation Register	32b0
Timer Interrupt Control and Status Register (TINT_CSTAT)				
<i>Address : TIMER : C0017044h / PWM : C0018044h</i>				
[31:10]	-	RESERVED	Reserved	-

Bit	R/W	Symbol	Description	Reset Value
[9]	R/W	TIMER 4 INTERRUPT STATUS	Timer 4 Interrupt Status Bit. Clears by writing '1' on this bit.	1'b0
[8]	R/W	TIMER 3 INTERRUPT STATUS	Timer 3 Interrupt Status Bit. Clears by writing '1' on this bit.	1'b0
[7]	R/W	TIMER 2 INTERRUPT STATUS	Timer 2 Interrupt Status Bit. Clears by writing '1' on this bit.	1'b0
[6]	R/W	TIMER 1 INTERRUPT STATUS	Timer 1 Interrupt Status Bit. Clears by writing '1' on this bit.	1'b0
[5]	R/W	TIMER 0 INTERRUPT STATUS	Timer 0 Interrupt Status Bit. Clears by writing '1' on this bit.	1'b0
[4]	R/W	TIMER 4 INTERRUPT ENABLE	Timer 4 Interrupt Enable. 1 : Enabled 0 : Disabled	1'b0
[3]	R/W	TIMER 3 INTERRUPT ENABLE	Timer 3 Interrupt Enable. 1 : Enabled 0 : Disabled	1'b0
[2]	R/W	TIMER 2 INTERRUPT ENABLE	Timer 2 Interrupt Enable. 1 : Enabled 0 : Disabled	1'b0
[1]	R/W	TIMER 1 INTERRUPT ENABLE	Timer 1 Interrupt Enable. 1 : Enabled 0 : Disabled	1'b0
[0]	R/W	TIMER 0 INTERRUPT ENABLE	Timer 0 Interrupt Enable. 1 : Enabled 0 : Disabled	1'b0

Section 20. Analog Digital Converter (ADC)

20.1 Overview

The ADC2802A is a 28nm CMOS 1.8V 12-bit analog-to-digital converter (ADC) with 16-ch analog input MUX and level-shifters for low-voltage digital interface. It converts single-ended analog input signal to 12-bit digital output codes at a maximum conversion rate of 1MSPS.

The device is a cyclic type monolithic ADC, which provides an on-chip sample-and-hold and power down mode.

20.1.1 Features

- 28nm Low Power CMOS Process
- Resolution: 12-bit
- Conversion rate (Fs): 1MSPS
- Power consumption:
 - 1.0 mW (Fs = 1MSPS) @ Normal operation mode Typ
 - 0.005 mW @ Power down mode Typ
- Input range: 0 ~ AVDD18(Normally 1.8V)
- Input frequency: up to 100kHz
- Digital output: CMOS Level (0 ~ AVDD10)
- Operation temperature range (ambient) : -25°C ~ 85°C

20.1.2 Block Diagram

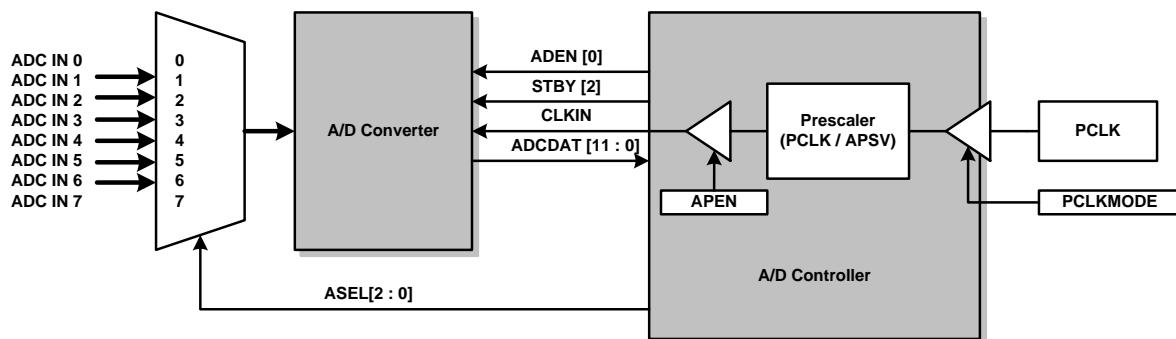


Figure 20-1. ADC Block Diagram

20.2 Functional Description

The NXP4330D/Q can receive eight ADC inputs and select one of the ADC inputs by using **ADCCON.ASEL[2:0]**. The sizes of the eight ADC inputs depend on the size of V_{REF}.

The ADC controller uses PCLK and the PCLK is divided in the Prescaler and applied to the ADC. At this time, the **ADCCON.APEN** bit is used for the application of CLKIN. Clock divide values by the Prescaler are available from 20 to 256. (Actually, since the register input value is [Clock Divide Value – 1], smaller divide values make the sampling more detailed. The clock divide value is determined by **ADCCON.APSV** bit.

If the ADC block continuously accepts after power is applied, it consumes current unnecessarily. In this case, it is better to power down the A/D converter by using the **ADCCON.STBY** bit. The **ADCCON.STBY** bit determines the power input for the ADC block. If the **ADCCON.STBY** bit is '0', the ADC block waits for ADC input after power on. After that, if the **ADCCON.ADEN** bit is set as '1' to accept ADC input, the ADC operation is actually performed. On the other hand, if the **ADCCON.STBY** bit is '1', the ADC block goes to power down status, that is, power is not applied. This is called Standby mode. (In Standby mode, only about 20 uA current is consumed.

If the ADC block is not used, set the **ADCCON.STBY** bit as '1' and power off the ADC block. In this way, unnecessary power consumption by the ADC block can be reduced. In addition, since the supply of the clock can be determined by using the **ADCCON.APEN** bit, power consumption can be reduced further by stopping the clock supply when supply is unnecessary.

20.2.1 I/O Chart

Index	ADC Input (V)	Digital Output	
0	~ 0.00322	00_0000_0000	1LSB = 3.22mV V _{REF} = 3.3V AGND = 0.0V
1	0.00322 ~ 0.00644	00_0000_0001	
2	0.00644 ~ 0.00967	00_0000_0010	
~	~	~	
511	1.64678 ~ 1.65000	01_1111_1111	
512	1.65000 ~ 1.65322	10_0000_0000	
513	1.65322 ~ 1.65644	10_0000_0001	
~	~	~	
1021	3.29033 ~ 3.29355	11_1111_1101	
1022	3.29355 ~ 3.29678	11_1111_1110	
1023	3.29678 ~	11_1111_1111	

Table 20-1. I/O Chart

20.2.2 Timing

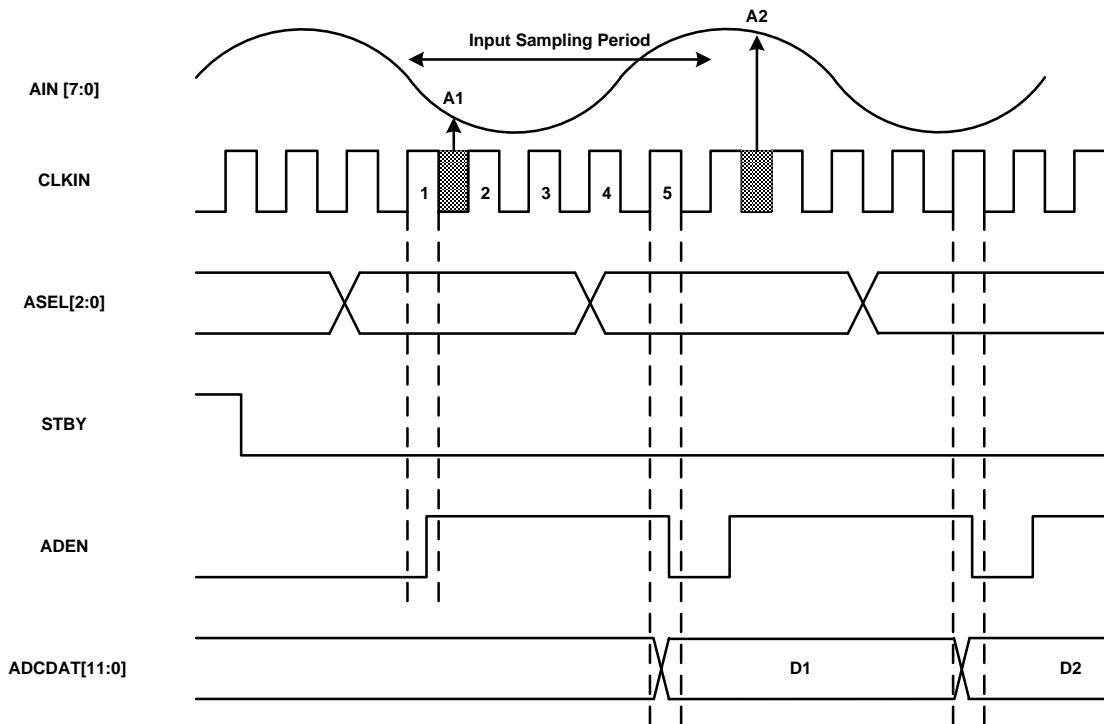


Figure 20-2. Main Waveform

Figure 20-2 shows the timing chart for the ADC. AIN[7:0] is continuously input from the outside and CLKIN is supplied via the **ADCCON.APEN** bit. After AIN[7:0] is selected, by using ASEL[2:0], the **ADCCON.STBY** bit is set as '0' to supply power to the ADC block. Finally, A/D conversion is progressed by setting the **ADCCON.ADEN** bit as '1'. After the conversion is completed, EDO occurs and the **ADCCON.ADEN** bit is automatically cleared to '0'. After that, A/D Converted Data (D1) can be read through **ADCDAT.ADCDAT**. Since it always takes 5-cycles for 10-bit conversion, the maximum conversion rate of the NXP4330D/Q is 1MSPS. Set the **ADCCON.ADEN** bit as '1' to operate the ADC again.

20.2.3 Analog Input Selection Table

CHANNEL	ASEL[2]	ASEL[1]	ASEL[0]
Analog Input [0]	0	0	0
Analog Input [1]	0	0	1
Analog Input [2]	0	1	0
Analog Input [3]	0	1	1
Analog Input [4]	1	0	0
Analog Input [5]	1	0	1
Analog Input [6]	1	1	0
Analog Input [7]	1	1	1

Table 20-2. Analog Input Selection Table

20.2.4 Flowchart

- PCLK Supply : *CLKENB.PCLKMODE* = 1
- Analog Input Select : *ADCCON.ASEL*
- ADC Power On : *ADCCON.STBY* = 0
- CLKIN Divide Value : *ADCCON.APSV*
- CLKIN On : *ADCCON.APEN*
- ADC Enable : *ADCCON.ADEN*
- A/D Conversion Process
- Read *ADCDAT.ADCDAT*
- CLKIN Off
- ADC Power Off

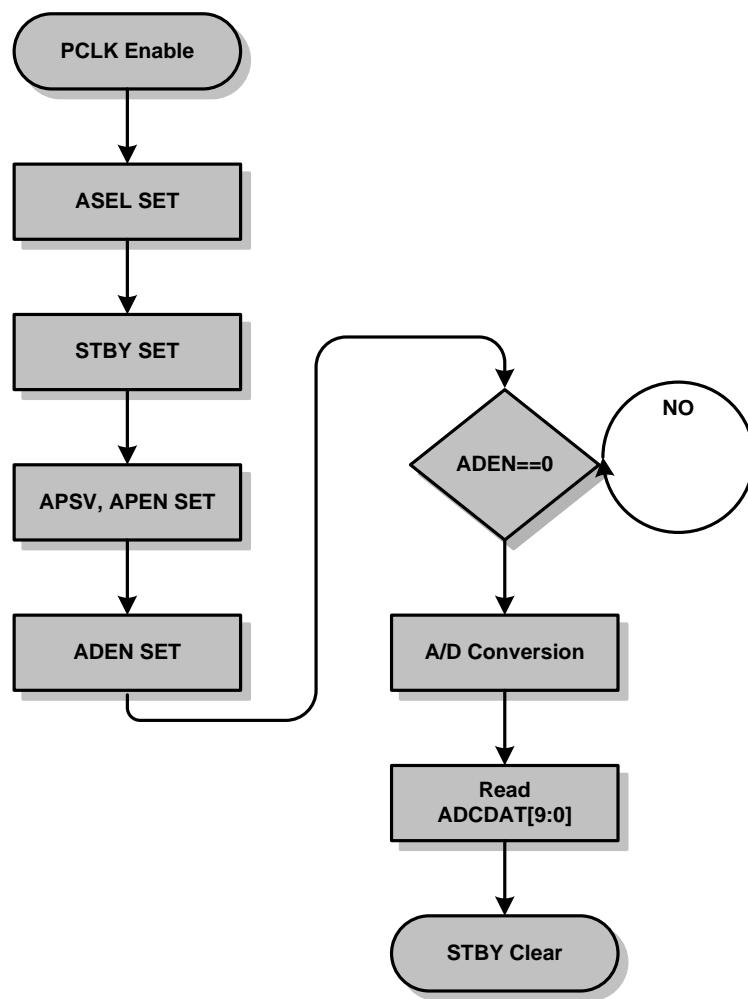


Figure 20-3. ADC Sequence Flowchart

20.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
ADC CONTROL REGISTER (ADCCON)				
Address : C0053000h				
[31:15]	R	RESERVED	Reserved	17'b1
[14]	R/W	APEN	Prescaler Enable.This bit determines the supply of the clock divided by the APSV register for the A/D converter.Before the APEN bit is enabled, the APSV register should be set. 0 : Disable 1 : Enable	1'b0
[13 : 6]	R/W	APSV	A/D Converter Clock Prescaler Value (8bit) - To write a value to this bit, APEN should be '0'. - The maximum value of the ADC CLK divided by the APSV value is 2.5 MHz (400 ns) (where PCLK is 50 MHz). - The minimum and the maximum value for the APSV bit are 19 and 255, respectively.(In effect, the range of the clock divide value is from 20 to 256). - Input Value = Clock Divide Value -1.For divide-by-20 and divide-by-100, (20-1) = 19 and (100-1) = 99 are input to APSV, respectively.	8'hFF
[5 : 3]	R/W	ASEL	These bits select ADCIN. NXP4330D/Q has four ADCINs and can select one of them. 000 : ADCIN_0 010 : ADCIN_2 100 : ADCIN_4 110 : ADCIN_6 001 : ADCIN_1 011 : ADCIN_3 101 : ADCIN_5 111 : ADCIN_7	3'b0
[2]	R/W	STBY	A/D Converter Standby Mode.If this bit is set as '0', power is actually applied to the A/D converter. 0 : ADC Power On 1 : ADC Power Off(Standby)	1'b1
[1]	-	RESERVED	Reserved	1'b0
[0]	R/W	ADEN	A/D Conversion Start - When the A/D conversion ends, this bit is cleared to '0'. Read > Check the A/D conversion operation. 0 : Idle 1 : Busy Write > Start the A/D conversion. 0 : None 1 : Start A/D conversion	1'b0
ADC OUTPUT DATA REGISTER (ADCDAT)				
Address : C0053004h				
[31 : 12]	-	RESERVED	Reserved	20'b0
[11 : 0]	R	ADCDAT	These bits are 12-bit data converted via the ADC.	12'b0
ADC INTERRUPT ENABLE REGISTER (ADCINTENB)				
Address : C0053008h				
[31 : 1]	-	RESERVED	Reserved	31'b0
[0]	R/W	ADCINTENB	ADC Interrupt Enable. (This bit determines the generation of an interrupt when EOC occurs.) This bit determines if interrupt occurs when the ADEN bit is '0'. 0 : Interrupt Disable 1 : Interrupt Enable	1'b0
ADC INTERRUPT PENDING AND CLEAR REGISTER (ADCINTCLR)				
Address : C005300Ch				
[31 : 1]	-	RESERVED	Reserved	31'b0

Bit	R/W	Symbol	Description	Reset Value
[0]	R/W	ADCINTCLR	EOC Interrupt Pending and Clear Read > 0 : None Write > 0 : None	1 : Interrupt Pended 1 : Pending Clear

Section 21. I2C Controller

21.1 Overview

The NXP4330D/Qapplication processor can support a multi-master I2C-bus serial interface. A dedicated serial data line (SDA) and a serial clock line (SCL) carry information between bus masters and peripheral devices which are connected to the I2C-bus. The SDA and SCL lines are bi-directional. Devices communicating with each other on a serial bus must have some form of protocol that avoids all possibilities of confusion, data loss and blockage of information. The system must not be dependent on the devices connected to it, otherwise modifications or improvements would be impossible.

In multi-master I2C-bus mode, multiple microprocessor can receive or transmit serial data to or from slave devices. The master that initiates a data transfer over the I2C-bus is responsible for terminating the transfer.

21.1.1 Features

- Compliance to the AMBA specification onwards for easy integration into SoC implementation
- Only 2 bus lines are required; a serial data line (SDA) and a serial clock line (SCL). The simple 2-wire serial I2C-bus minimizes interconnections so ICs have fewer pins.
- The completely integrated I2C-bus protocol eliminates the need for address decoders and other glue logic.
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; master can operate as master-transmitter or as master-receiver.
- It is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer.
- Serial, 8-bit oriented, bi-directional data transfers can be made at up to 100kbit/s in the Standard-mode, up to 400kbit/s in the Fast-mode
- The number of ICs that can be connected to the same bus is limited only by a maximum bus capacitance of 400pF.
- System bus asynchronous reset is not supported.
- Every operation command used in I2C-bus needs delays of minimum 3-Cycles (PCLK) between them.
- High speed mode, combined format, 10bit address are not supported

21.2 Functional Description

21.2.1 The Concept of the I2C-Bus

The I2C-bus interface has four operation modes:

- Master transmitter mode
- Master receiver mode
- Slave transmitter mode
- Slave receiver mode

Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the bus. Each device is recognized by a unique address (whether it's a microcontroller, LCD driver, memory or keyboard interface) and operate as either a transmitter or receiver, depending on the function of the device. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfer. A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave. Table 21-1 shows the basic terminology of I2C-bus.

Term	Definition
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Term	Definition
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Multi-Master	More than one master can attempt to control the bus at the same time without corrupting the message
Slave	The device addressed by a master
Arbitration	Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the winning message is not corrupted
Synchronization	Procedure to synchronize the clock signals of two or more devices

Table 21-1 I/O I2C-Bus terminology definition

21.2.2 IC Protocol

Both SDA and SCL are bi-directional lines, connected to a positive supply voltage via a current-source or pull-up resistor. Figure 21-1 shows the hardware layout of the I2C-bus. When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function. Data on the I2C-bus can be transferred at rates of up to 100kbit/s in the Standard-mode or up to 400kbit/s in the Fast-mode.

Figure 21-2 shows additional information of I2C-bus pad structure.

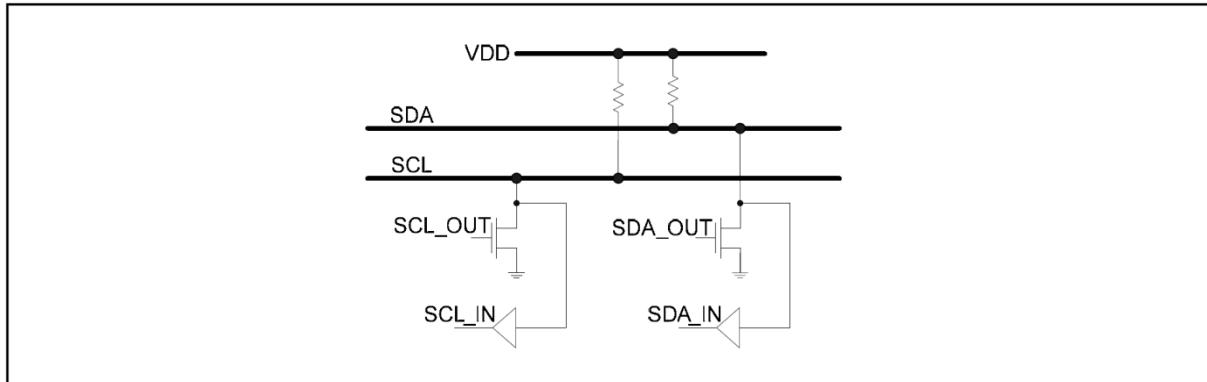


Figure 21-1 Connection of devices to the I2C-Bus

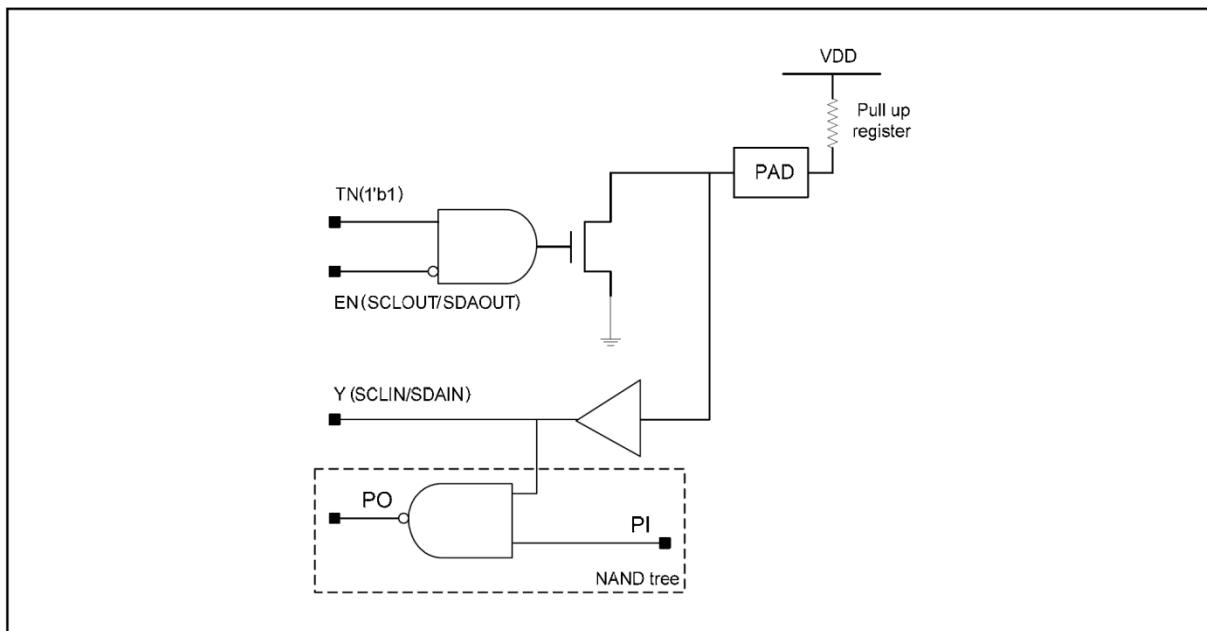


Figure 21-2 Bi-direction PAD structure of the I2C-Bus

21.2.3 Start/Stop operation

Within the procedure of the I2C-bus, unique situations arise which are defined as START (S) and STOP (P) condition. A HIGH to LOW transition on the SDA line while SCL is HIGH is one such unique case. This situation indicates a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP condition.

START and STOP condition are always generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

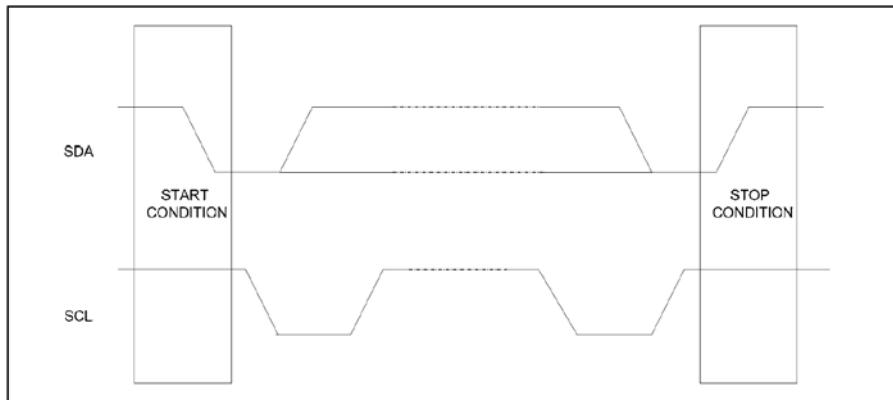


Figure 21-3 Start/Stop condition of I2C-Bus

21.2.4 Data Format

The first seven bits of the first byte make up the slave address. The eighth bit is the LSB (Least Significant Bit). It determines the direction of the message. A 'zero' in the least significant position of the first byte means that the master will write information to a selected slave. A 'one' in this position means that the master will read information from the slave. A data transfer is always terminated by a STOP condition (P) generated by the master. The first acknowledge is generated by the slave. But the acknowledge of data is generated by the receiver. Figure 21-4 shows the 7-bit data format.

Possible data transfer formats are:

- Master-transfer transmits to slave receiver.
- Master reads slave immediately after first byte. At the moment of the first acknowledge, the master-transfer becomes a master-receiver and the slave-receiver becomes a slave-transfer.
- Combined format is not supported.

NOTE: A START condition immediately followed by a STOP condition (void message) is an illegal format.

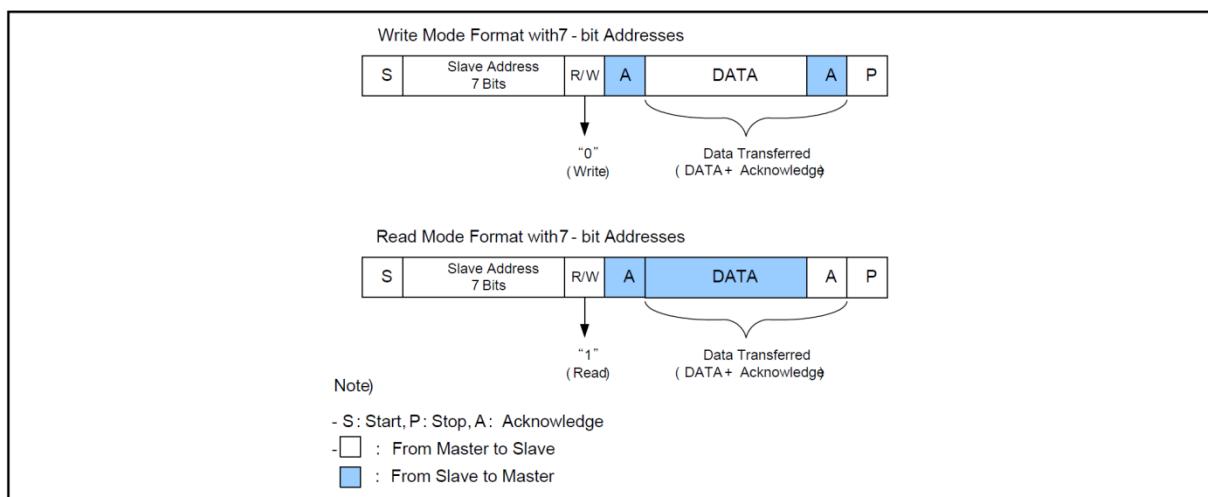


Figure 21-4 I2C-Bus data format

21.2.5 Data Transfer

Every byte put on the SDA line must be eight bits long. The number of bytes which can be transmitted per transfer is

unrestricted. The address byte is transmitted by the master when the I2C-bus is operating in master mode. When a master initiates a start condition, it sends its slave address onto the bus. The address byte consists of a 7-bit address and a 1-bit transfer direction indicator (that is, write or read). If bit 8 is "0", a transmit operation (write) is indicated; if bit 8 is "1", a request for data (read) is indicated.

The master ends the indicated transfer operation by transmitting a stop condition. If the master wants to continue sending data over the bus, it can generate another start condition and another slave address. Each byte must be followed by an acknowledge (ACK) bit. Serial data and addresses are transferred with the most significant bit (MSB) first. In this way, read-write operations can be performed in various formats. Figure 21-5 shows the sequence of data transmission.

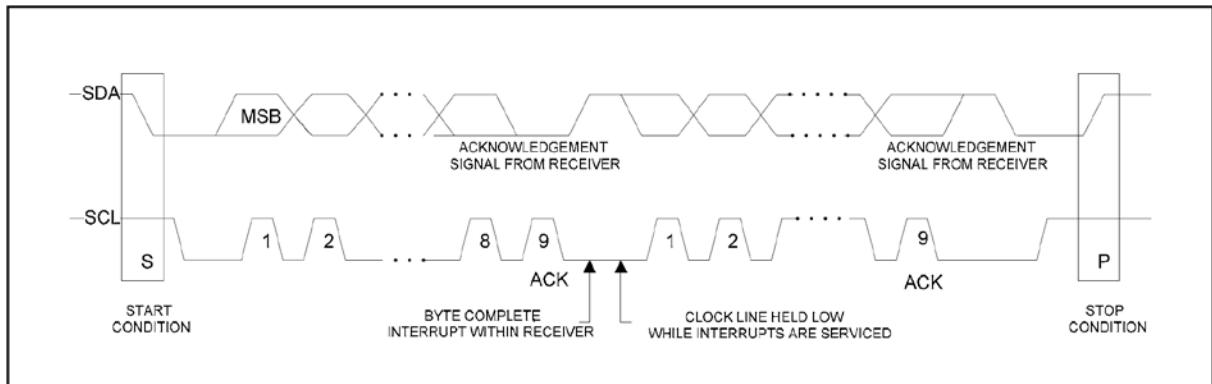


Figure 21-5 Data transfer on the I2C-Bus

21.2.6 Arbitration

Arbitration takes place on the SDA line to prevent contention on the bus between two masters, while the SCL line is at the HIGH level. If a master with a SDA High level detects another master with a SDA Low level, it will not initiate a data transfer because the current level on the bus does not correspond to its own. The master that loses the arbitration can generate clock pulses until the end of the last-transmitted data byte. The arbitration procedure can continue while data continues to be transferred over the bus. Figure 21-6 shows the arbitration.

The first stage of arbitration is the comparison of address bits. If a master loses the arbitration during the addressing stage of a data transfer, it is possible that the master which won the arbitration is attempting to address the master which lost. In this case, the losing master must immediately switch to the slave receiver mode.

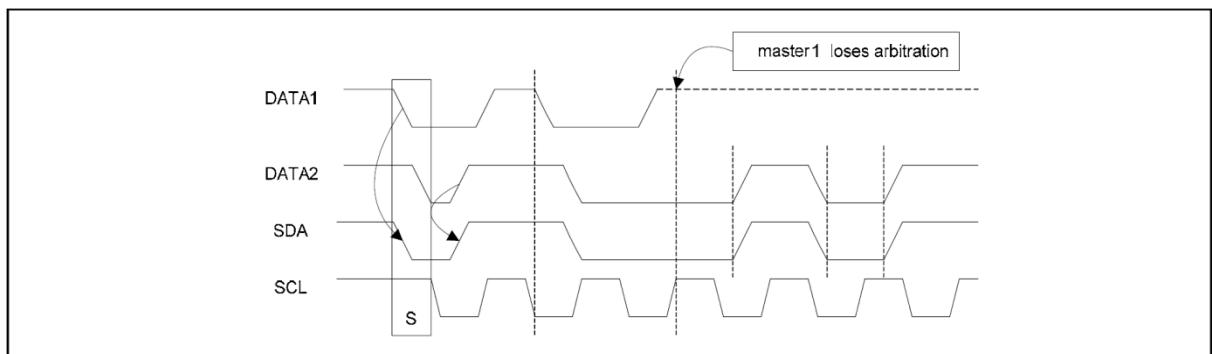


Figure 21-6 Arbitration procedure between two masters

21.2.7 Synchronization

Clock synchronization is performed using the wired-AND connection of I2C-bus interface to the SCL line. This means

that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and, once a device clock has gone LOW, it will hold SCL line in that state until the clock HIGH state is reached. However, the LOW to HIGH transition of this clock may not change the state of the SCL line if another clock is still within its LOW period. In this way, a synchronized SCL clock is generated with its LOW period determined by the device with the longest clock LOW period and its HIGH period determined by the one with the shortest clock HIGH period. Devices with shorter LOW periods enter a HIGH wait-state during this time. Figure 21-7 shows the procedure of SCL generation.

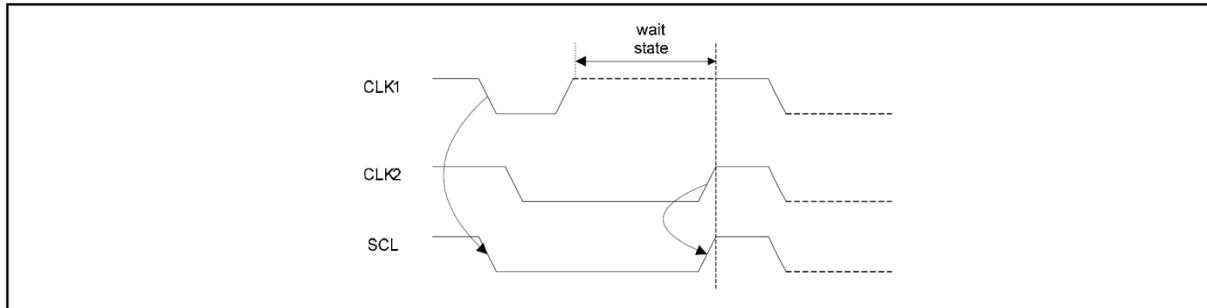


Figure 21-7 Clock synchronization

21.2.8 Acknowledge

Data transfer with acknowledge is obligatory. To complete one-byte transfer operation, the receiver must send an ACK bit to the transmitter. The ACK pulse occurs at the ninth clock of the SCL line (eight clocks are required to complete the one-byte transfer).

The transmitter releases the SDA line (that is, it sends the SDA line High) when the ACK clock pulse is received. The receiver must drive the SDA line Low during the ACK clock pulse so that SDA is Low during the High period of the ninth SCL pulse.

The ACK bit transmit function can be enable and disabled by software (ICCR[7]). Figure 21-8 shows data transfer with acknowledge signal.

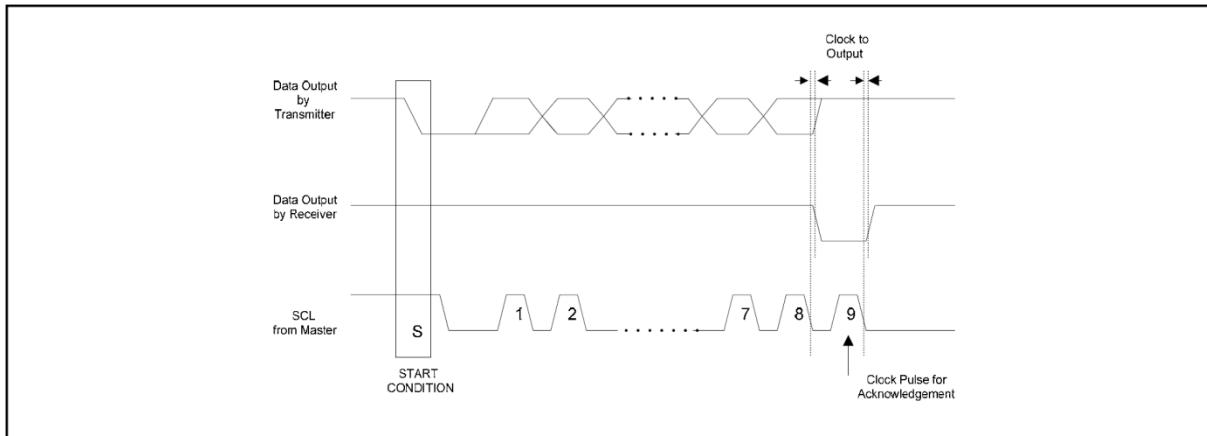


Figure 21-8 Acknowledge on the I2C-Bus

21.2.9 Read/Write operation

When operating in transmitter mode, the I2C-bus interface interrupt routine waits for the master to write a data byte into the I2C-bus data shift register (IDSR). To do this, it holds the SCL line Low prior to transmission. In receive mode, the I2C-bus interface waits for the master to read the byte from the I2C-bus data shift register (IDSR). It does this by holding

the SCL line Low following the complete reception of a data byte.

21.2.10 Configuration I2C-Bus

Data transfer with acknowledge is obligatory. To complete one-byte transfer operation, the receiver must send an ACK bit to the transmitter. The ACK to control the frequency of the serial clock (SCL), user has to program the 4-bit prescaler value in the ICCR register.

Timing of the SCL and SDA

- Standard mode: $f_{SCL} = \text{MAX. } 100\text{kHz}$ (Period: 10us)
- Fast mode : $f_{SCL} = \text{MAX. } 400\text{kHz}$ (period:2.5us)

Parameter	Symbol	Standard mode		Fast mode		Unit
		MIN	MAX	MIN	MAX	
SCL clock frequency	f_{SD}	0	100	0	400	kHz
Hold time Start condition. After this period, the first clock pulse is generated	$t_{holdstart}$	4.0	-	0.6	-	μs
Low period of SCL clock	t_{LOW}	4.7	-	1.3	-	μs
High period of SCL clock	t_{HIGH}	4.0	-	0.6	-	μs
Data hold time	$t_{hold:DATA}$	0	3.45	0	0.9	μs
Data setup time	$t_{setup:DATA}$	250	-	100	-	ns
Rise time of both SDA and SCL signals	t_r	-	1000	$20+0.1C_b$	300	ns
Fall time of both SDA and SCL signals	t_f	-	300	$20+0.1C_b$	300	ns
Setup time for Stop condition	$t_{setup:STOP}$	4.0	-	0.6	-	μs
Captive load for each bus line	C_b	-	400	-	400	pF
Bus free time between a Stop and Start condition	$t_{BUSfree}$	4.7	-	1.3	-	μs

Table 21-2 Characteristics of the SDA and SCL bus lines for Standard and Fast mode I2S-bus

NOTE:

- C_b is total capacitance of one bus line in pF
- Timing spec for SCL dose not affect slave mode devices. (EG. High/Low SCL period, Start/Stop data hold time)

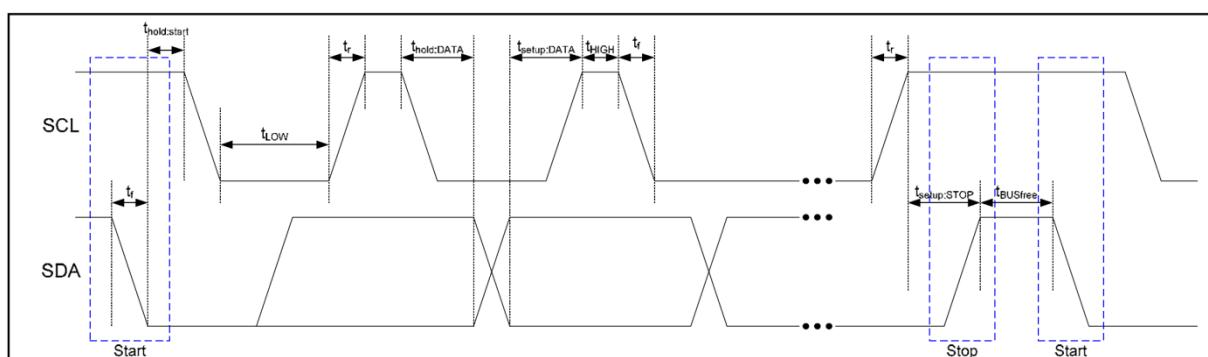


Figure 21-9 Timing on the SCL and SDA

21.2.11 Operations

To control multi-master I2C-bus operations, user has to write values to the following registers:

- Multi-master I2C-bus control register, ICCR
- Multi-master I2C-bus control-status register, ICSR
- Multi-master I2C-bus Tx/Rx data shift register, IDSR
- Multi-master I2C-bus address register, IAR
- Multi-master I2C-bus stop control register, STOPCON

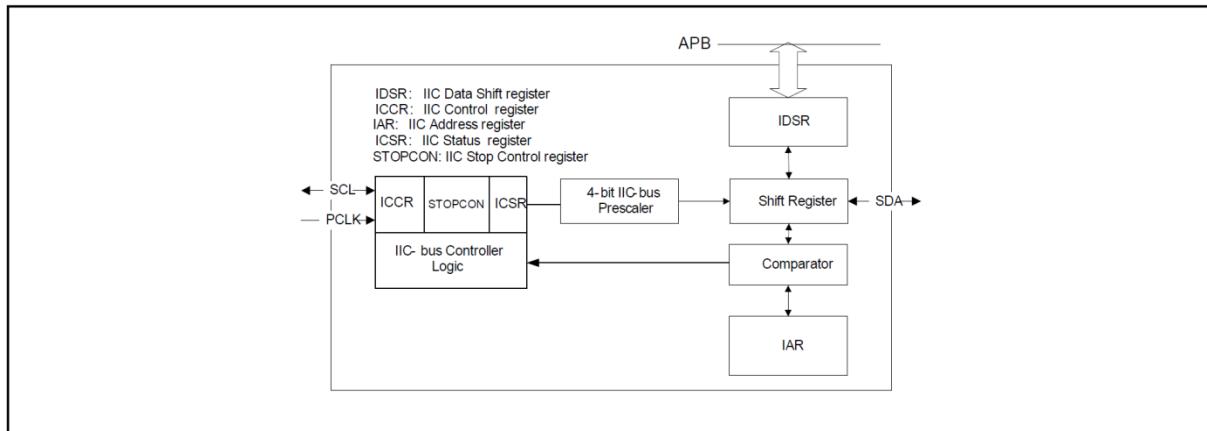


Figure 21-10 Functional block diagram of I2C-Bus

21.2.12 Interrupt generation

When one byte transmission or reception, appointed as slave and a loss of the bus arbitration, this block requests interrupt to external interrupt controller. To enable the I2C-bus interrupt, the ICCR[5] bit should be set to '1'. When the I2C-bus interrupt is enabled, the interrupt signal generates request signal to CPU.

21.2.13 System bus reset

The I2C-bus was designed for system bus clock synchronous reset scheme. Asynchronous reset scheme is forbidden.

21.3 Programming Guide

21.3.1 I2C-bus Initialize

The initialization sequence in this section is used for I2C-bus controller.

- 1) Provide a clock and Release a reset This signals are propagated all of the I2C-bus controller internal registers and logic.
- 2) Enable pad select alt-function of I2C

21.3.2 Commands

Following diagram shows I2C-bus operation command examples.

21.3.2.1 Master Transmitter Mode (M/Tx)

In this mode, master-transmitter addresses slave, sends data to slave-receiver, and terminates the transfer.

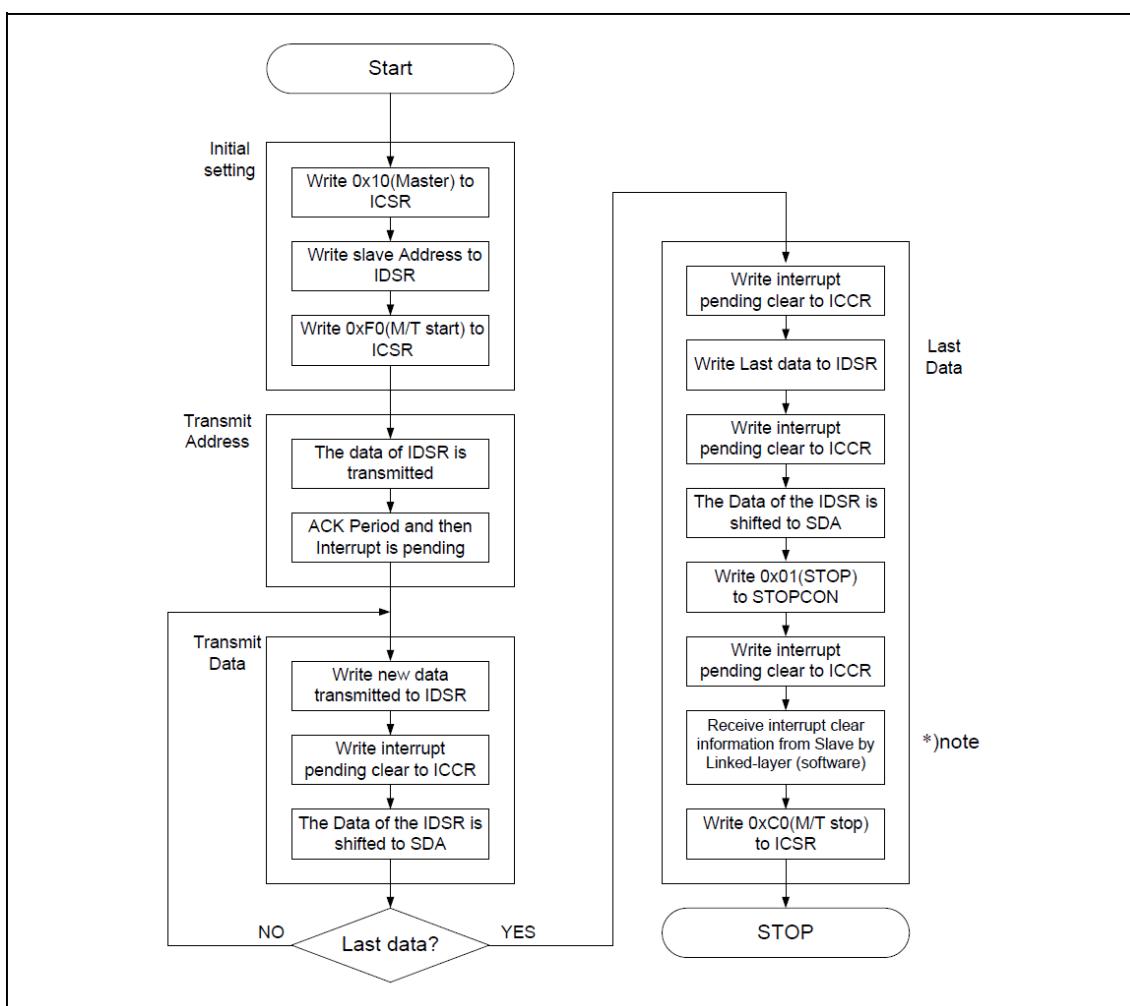


Figure 21-11 Master transmitter mode operation

21.3.2.2 Master Receiver Mode (M/Rx)

In this mode, master-receiver addresses slave, receives data from slave-transmitter, and terminates the transfer.

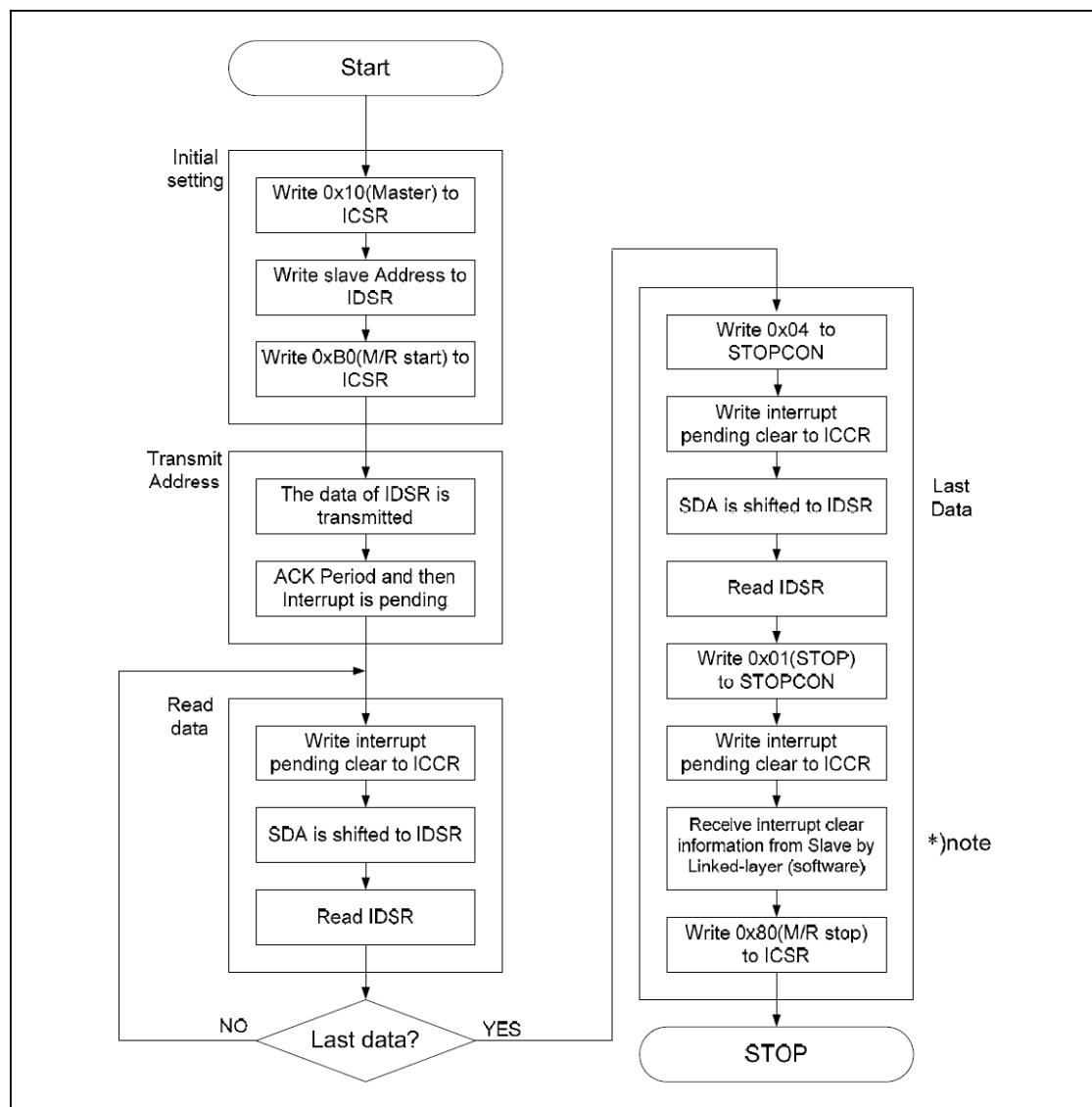


Figure 21-12 Master receiver mode operation

21.3.2.3 Slave Transmitter Mode (S/Tx)

In this mode, received address is compared with IAR register value. If the current I2C-bus interface selected, slave-transmitter transfers data to master-receiver.

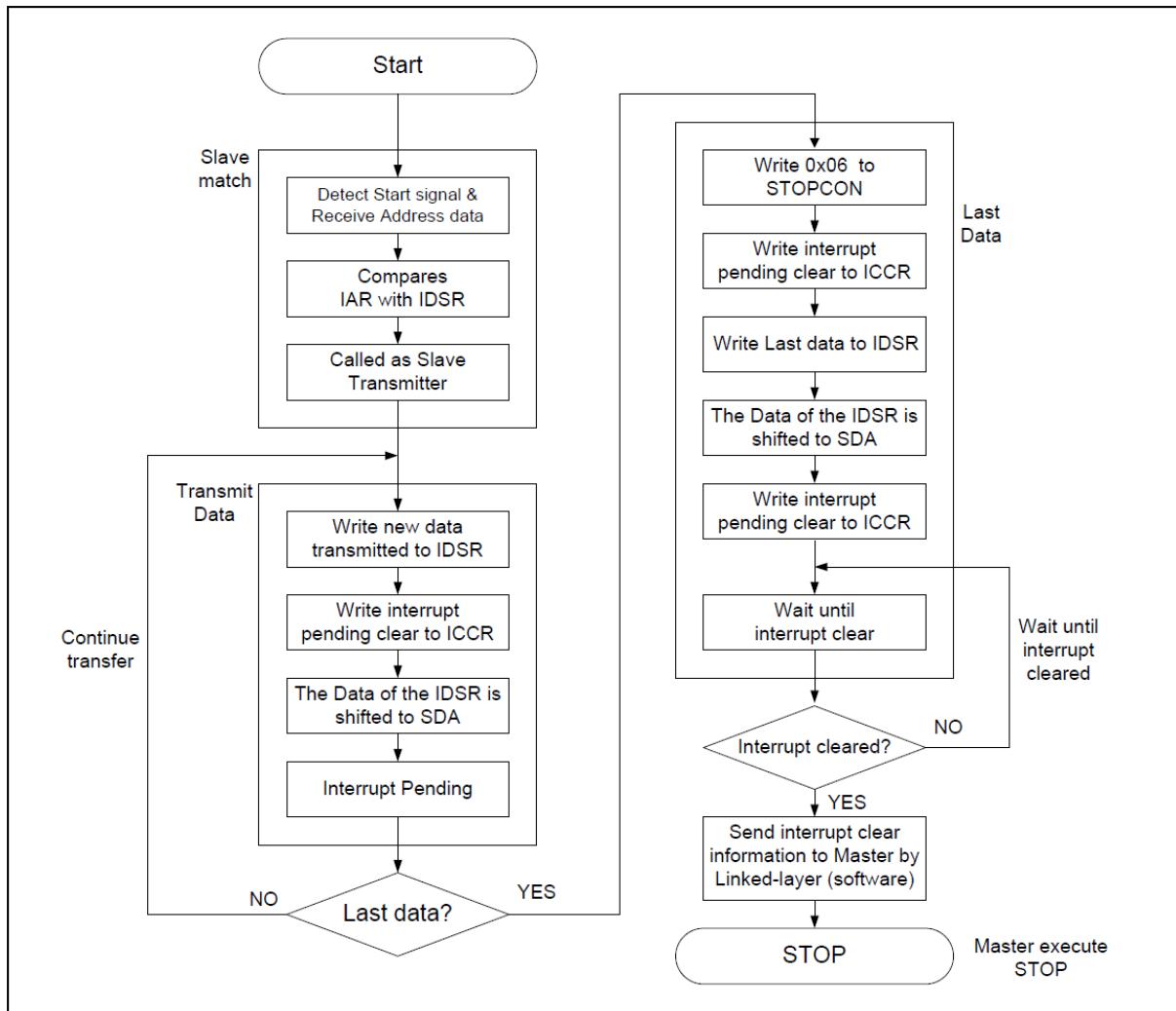


Figure 21-13 Slave transmitter mode operation

21.3.2.4 Slave Receiver Mode (S/Rx)

In this mode, received address is compared with IAR register value. If the current I2C-bus interface selected, slave-receiver receives data from master-transmitter.

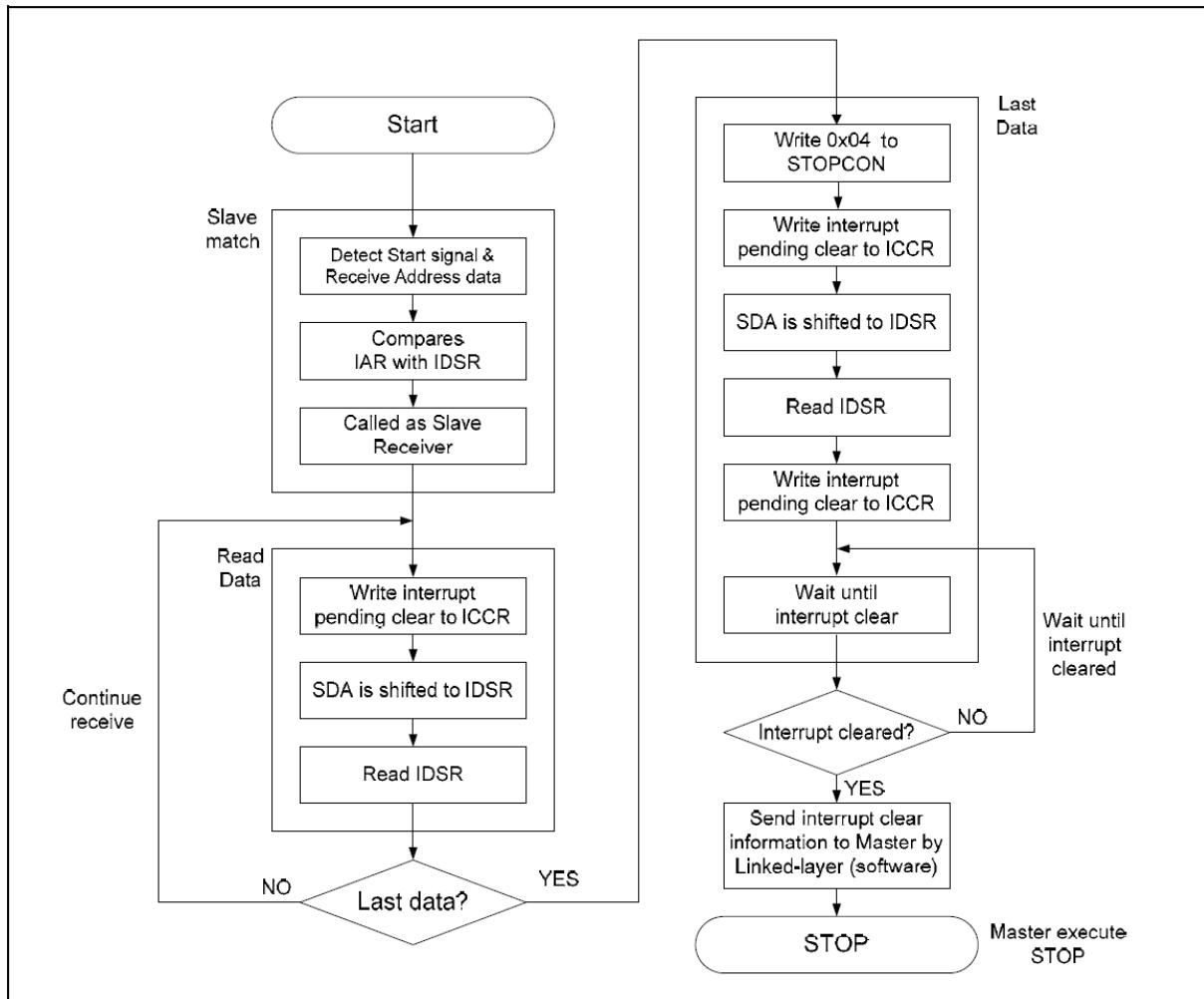


Figure 21-14 Slave receiver mode operaiton

21.4 Design Issues

21.4.1 Software Layer

This I2C-bus module needs linked-layer (software) to control both module operations. For example, STOP sequence of I2C-bus operation needs the status of both I2C-bus modules. Figure 21-15 shows a simple example how to deal with a STOP sequence of I2C-bus by linked-layer (by software).

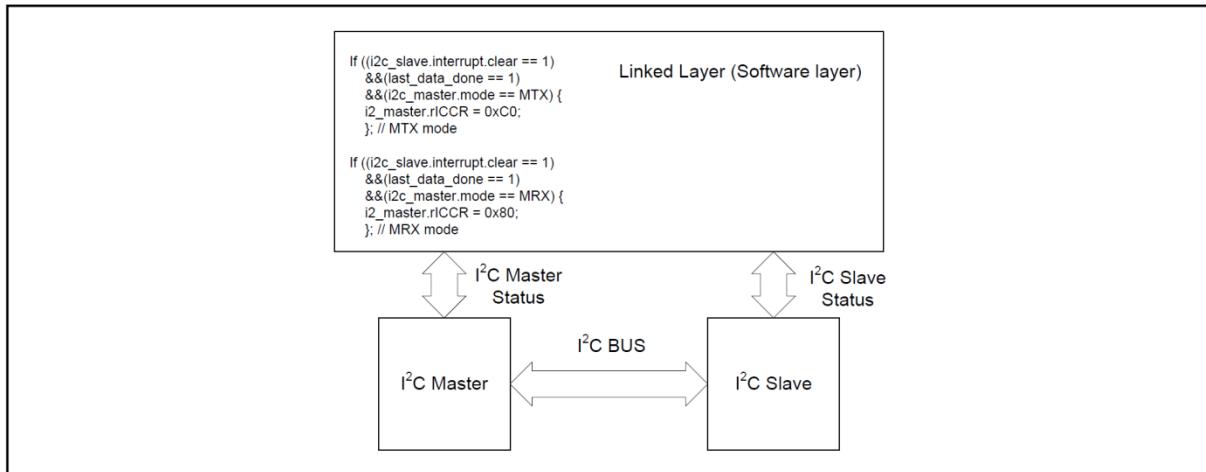


Figure 21-15 Linked layer of the I2C-bus

21.4.2 Delays between I2C-bus commands

This I2C-bus module needs 3-cycle delays between every register setting operations. For example, delay operation must be inserted between the Tx data write command (IDSR) and the interrupt clear command (ICCR).

Figure 21-16 shows a simple example how to deal with delays between each register setting operation.

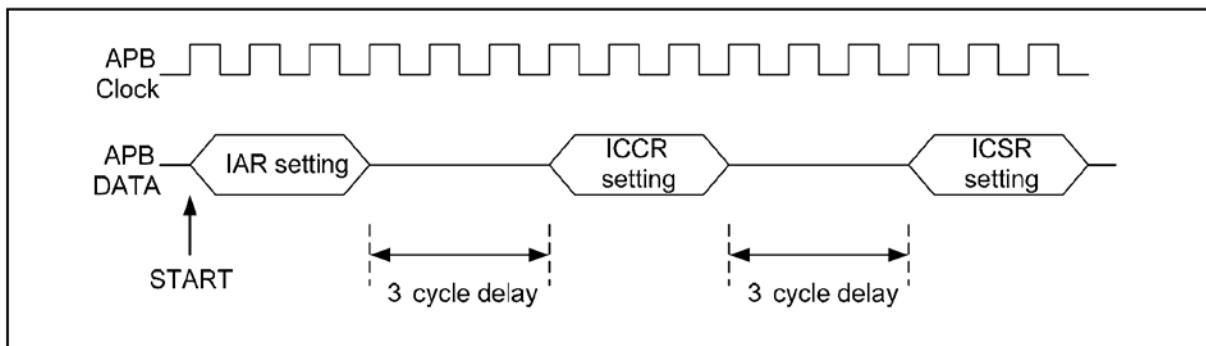


Figure 21-16 Delays between the I2C-bus commands

21.5 Register Summary

Bit	R/W	Symbol	Description	Reset Value
I2C-bus control register (ICCR) Address : C00A_4000h (I2C 0) Address : C00A_5000h (I2C 1) Address : C00A_6000h (I2C 2)				
[8]	R/W	INTERRUPT CLEAR	I2C-bus interrupt clear bit 0: No interrupt (when read) 1: Interrupt is cleared (when write)	1'b0
[7]	R/W	ACKNOWLEDGE ENABLE	I2C-bus acknowledge enable bit 0: Disable ACK generation 1: Enable ACK generation	1'b0
[6]	R/W	TX CLOCK SOURCE SELECTION	Source clock of I2C-bus transmit clock prescaler selection bit 0: I2C CCLK = $f_{\text{CLK}}/16$ 1: I2C CCLK = $f_{\text{CLK}}/256$	1'b0
[5]	R/W	TX/RX INTERRUPT ENALBE	I2C-bus Tx/Rx interrupt enable/disable bit 0: Disable interrupt 1: Enable interrupt	1'b0
[4]	R/W	INTERRUPT PENDING FLAG	I2C-bus Tx/Rx interrupt pending flag. Writing "1" is impossible 0: No interrupt pending (when read), This bit is cleared (when write) 1: Interrupt is pending (when read), No effect. Namely '1' doesn't be written to this bit <NOTE> A I2C-bus interrupt occurs 1) When an 1-byte transmit or receive operation is terminated 2) When a general call or a slave address match occurs 3) If bus arbitration fails	1'b0
[3:0]	R/W	TRANSMIT CLOCK VALUE	I2C-bus transmit clock prescaler. I2C-bus transmit clock frequency is determined by this 4-bit prescaler value, according to the following formula: $Tx\ clock = I2C\ CLK / (ICCR[3:0]+1)$ <NOTE> 1) I2CCLK is determined by ICCR[6] 2) Tx clock can vary by SCL transition time 3) When ICCR[6] = 0, "ICCR[3:0] = 0x0 or 0x1" is not available	4'bXXXX
I2C-bus Control-Status Register (ICSR) Address : C00A_4004h (I2C 0) Address : C00A_5004h (I2C 1) Address : C00A_6004h (I2C 2)				
[7:6]	R/W	MODE SELECTION	I2C-bus master-slave Tx/Rx mode select bits: 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode	2b0
[5]	R/W	BUSY SIGNAL STATUS	I2C-bus busy signal status bit: 0: I2C-bus not busy (when read), I2C-bus interface STOP signal generation (when write) 1: I2C-bus busy (when read), I2C-bus interface START signal generation (when write)	1'b0
[4]	R/W	SERAIL OUTPUT ENABLE	I2C-bus data output enable/disable bit: 0: Disable Rx/Tx 1: Enable Rx/Tx	1'b0
[3]	R	ARBITRATION STATUS FLAG	I2C-bus arbitration procedure status flag bit 0: Bus arbitration status okay 1: Bus arbitration failed during serial I/O	1'b0
[2]	R	ADDRESS-AS-SLAVE STATUS FLAG	I2C-bus address-as-slave status flag bit 0: START/STOP condition was generated	1'b0

Bit	R/W	Symbol	Description	Reset Value
			1: Received slave address matches the address value in the IAR	
[1]	R	ADDRESS ZERO STATUS FLAG	I2C-bus address zero status flag bit: 0: START/STOP condition was generated 1: Received slave address is '0x00'	1'b0
[0]	R	LAST-RECEIVED BIT STATUS FLAG	I2C-bus last-received bit status flag bit 0: Last-received bit si '0' (ACK was received) 1: Last-received bit is '1' (ACK was not received)	1'b0
I2C-bus Address Register (IAR)				
Address : C00A_4008h (I2C 0) Address : C00A_5008h (I2C 1) Address : C00A_6008h (I2C 2)				
[7:1]	R/W	SLAVE ADDRESS	7-bit slave address, latched from the I2C-bus: When serial output enable='0' in the ICSR, IAR is write-enable. You can read the IAR value at any time, regardless of the current serial output enable bit (ICSR[4]) setting - Slave address = [7:1] - Not mapped = [0]	7'hXX
I2C-bus Transmit-Receive Data Shift Register (IDSR)				
Address : C00A_400Ch (I2C 0) Address : C00A_500Ch (I2C 1) Address : C00A_600Ch (I2C 2)				
[7:0]	R/W	DATA SHIFT	8-bit data shift register for I2C-bus Tx/Rx operation: When serial output enable='1' in the ICSR, IDSR is write-enabled. You can read the IDSR value at any time, regardless of the current serial output enable bit (ICSR[4]) setting	8'hXX
I2C-bus Stop Control Register (STOPCON)				
Address : C00A_4010h (I2C 0) Address : C00A_5010h (I2C 1) Address : C00A_6010h (I2C 2)				
[2]	R/W	NOT-ACKNOWLEDGE GENERATION DATA SHIFT CONTROL	I2C-not acknowledge generation and data shift control bit 0: Normal operation 1: Master will generate not-acknowledge for the last byte received in master-receive mode. Or I2C will not shift the data shift register IDSR after the stop.	1'b0
[1]	R/W	DATA LINE RELEASE	I2C-data bus release setting bit 0: Normal operation 1: Slave will release data bus line in Transfer mode after transmitting the last byte. <NOTE> This bit can be only used in slave-transmit mode	1'b0
[0]	R/W	CLOCK LINE RELEASE	I2C-clock bus release setting bit 0: Normal operation 1: Master will release clock bus line for the stop condition after completing the transfer. <NOTE> This bit can be only used in Master-transmit mode or Master-receive mode.	1'b0
<Note0> Writing 0x00 or 0x03 or 0x07 on the STOPCON register are not permitted				

Section 22. **SPI/SSP**

22.1 Overview

The SPI/SSP is a full-duplex synchronous serial interface. It supports Serial Peripheral Interface (SPI) and Synchronous Serial Protocol (SSP). It can connect to a variety of external converter, serial memory and many other device which use serial protocols for transferring data.

There are 4 I/O pin signals associated with SPI/SSP transfers: the SSPCLK, the SSPrxD data receive line, the SSPTxD data transfer line, SSPFSS (Chip Select in SPI mode , Frame Indicator in SSP mode).

The NXP4330D/Q has three SPI/SSP port and it can operate in Master and Slave mode.

22.1.1 Features

- SPI Protocol, SSP Protocol, Microwire Protocol
- 16-bit wide, 8-location deep transmit/receive FIFO
- Master & Slave mode
- DMA request servicing of the transmit and receive FIFO
- Inform the system that a receive FIFO over-run has occurred
- Inform the system that data is present in the receive FIFO after an idle period has expired
- Programmable clock bit rate and prescale
- Max Operation Frequency
 - Master Mode : 50MHz (Receive Data is 20 MHz)
 - Slave Mode : 8MHz

22.1.2 Block Diagram

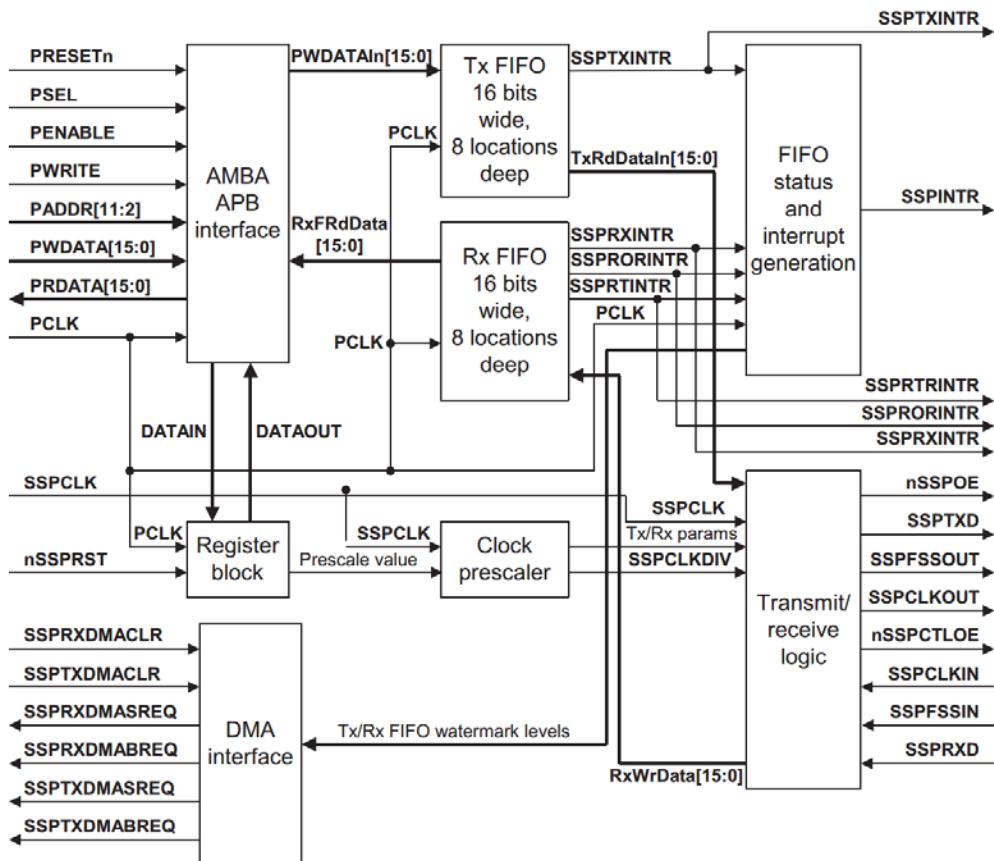


Figure 22-1. SPI/SSP Block Diagram

22.2 Functional Description

22.2.1 Clock Generation Configuration

SPI/SSP operates by using the PCLK and the SSPCLK. The PCLK is used when the CPU accesses the registers of the SPI/SSP. And, users can adjust the PCLK for the SPI/SSP by setting the PCLKMODE parameters according to their purpose. The PCLK is generated by Clock Generator of the SPI/SSP.

The SSPCLK is used when the SPI/SSP transmit or receive SPI/SSP Protocol. SSPCLK is generated by Clock Generator of the SPI/SSP. Each SPI/SSP has own Clock Generator. Therefore, users must set up the SPI/SSP Clock Generator before the SPI configuration stage.

22.2.1.1 Clock ratios

There is a constraint on the ratio of the frequencies of PCLK to SSPCLK. The frequency of SSPCLK must be less than or equal to that of PCLK. This ensures that control signals from the SSPCLK domain to the PCLK domain are certain to get synchronized before one frame duration:

$$F_{SSPCLK} \leq F_{PCLK}. (F_{SSPCLK} \leq 100MHz)$$

In the slave mode of operation, the SSPCLKIN signal from the external master is double synchronized and then delayed to detect an edge. It takes three SSPCLKs to detect an edge on SSPCLKIN. SSPTXD has less setup time to the falling edge of SSPCLKIN on which the master is sampling the line. The setup and hold times on SSPRXD with reference to SSPCLKIN must be more conservative to ensure that it is at the right value when the actual sampling occurs within the SSPMS. To ensure correct device operation, SSPCLK must be at least 12 times faster than the maximum expected frequency of SSPCLKIN.

The frequency selected for SSPCLK must accommodate the desired range of bit clock rates. The ratio of minimum SSPCLK frequency to SSPCLKOUT maximum frequency in the case of the slave mode is 12 and for the master mode it is two.

To generate a maximum bit rate of 1.8432Mbps in the Master mode, the frequency of SSPCLK must be at least 3.6864MHz. With an SSPCLK frequency of 3.6864MHz, the SSPCPSR register has to be programmed with a value of two and the SCR[7:0] field in the SSPCR0 register needs to be programmed as zero.

To work with a maximum bit rate of 1.8432Mbps in the slave mode, the frequency of SSPCLK must be at least 22.12MHz. With an SSPCLK frequency of 22.12MHz, the SSPCPSR register can be programmed with a value of 12 and the SCR[7:0] field in the SSPCR0 register can be programmed as zero. Similarly the ratio of SSPCLK maximum frequency to SSPCLKOUT minimum frequency is 254 x 256.

The minimum frequency of SSPCLK is governed by the following equations, both of which have to be satisfied:

$$F_{SSPCLK(min)} \Rightarrow 2 \times F_{SSPCLKOUT(max)} \text{ [for master mode]}$$

$$F_{SSPCLK(min)} \Rightarrow 12 \times F_{SSPCLKIN(max)} \text{ [for slave mode].}$$

The maximum frequency of SSPCLK is governed by the following equations, both of which have to be satisfied:

$$F_{SSPCLK(max)} \leq 254 \times 256 \times F_{SSPCLKOUT(min)} \text{ [for master mode]}$$

$$F_{SSPCLK(max)} \leq 254 \times 256 \times F_{SSPCLKIN(min)} \text{ [for slave mode]}$$

22.2.1.2 Programming the serial clock rate

SSPCR0 register is used to

- Program the serial clock rate
 - Select one of the three protocols
 - Select the data word size (where applicable)
- SSPCPSR register is used to
- Program the serial clock prescale divisor

The serial bit rate is derived by dividing down the input clock SSPCLK. The clock is first divided by an even prescale value CPSDVSR from 2 to 254, which is programmed in SSPCPSR. The clock is further divided by a value from 1 to 256, which is 1 + SCR, where SCR is the value programmed in SSPCR0.

The frequency of the output signal bit clock SSPCLKOUT is defined below:

$$F_{\text{SSPCLKOUT}} = \frac{F_{\text{SSPCLK}}}{\text{CPSDVR} \times (1+\text{SCR})}$$

For example, if SSPCLK is 3.6864MHz, and CPSDVSR = 2, then SSPCLKOUT has a frequency range from 7.2kHz to 1.8432MHz.

22.2.2 Operation

The SPI/SSP Block transfers Serial Data from / to External device via FIFO in the SPI/SSP Block. The transfer operation is initiated by CPU, using the programmed I/O system or the DMA, to / from the system memory. The SPI/SSP Data transfer is performed in full duplex.

When the SPI/SSP sends Data to the PIO mode, the transfer operation is completed by Reading 'Read FIFO' or Writing to 'Write FIFO' by means of a program.

In the case of communication with DMA mode, For DMA transfer mode, user must set up the DMA configuration and set as '1' the ***SSPDMA.RXDMAE*** bit and the ***SSPDMA.RXDMAE*** bit in the SPI/SSP to enable the DMA transfer Request from SPI/SSP. RX Request occurs when Receive FIFO has 4 more than data, TX Request occurs when Transmit FIFO has 4 less than data. User can't adjust this configuration.

22.2.2.1 Transmit and Receive operation

When configured as a master, the clock to the attached slaves is derived from a divided down version of SSPCLK through the prescaler operations described previously. The master transmit logic successively reads a value from its transmit FIFO and performs

parallel to serial conversion on it. Then the serial data stream and frame control signal, synchronized to SSPCLKOUT, are output through the SSPTXD pin to the attached slaves. The master receive logic performs serial to parallel conversion on the incoming

synchronous SSPRXD data stream, extracting and storing values into its receive FIFO, for subsequent reading through the APB interface.

Note : SSPRXD data is only valid when FSSPCLKOUT (Frequency of SSPCLKOUT) is lower than 20 MHz. This Constraint may be changed by exterior device's specification.

When configured as a slave, the SSPCLKIN clock is provided by an attached master and used to time its transmission and reception sequences. The slave transmit logic, under control of the master clock, successively reads a value from its transmit FIFO, performs parallel to serial conversion, then output the serial data stream and frame control signal

through the slave SSPTXD pin. The slave receive logic performs serial to parallel conversion on the incoming SSPRXD data stream, extracting and storing values into its receive FIFO, for subsequent reading through the APB interface.

22.2.2.2 Reset operation

The SPI/SSP is reset by the global reset signal PRESETn and a block-specific reset signal nSSPRST. An external reset controller must use PRESETn to assert nSSPRST asynchronously and negate it synchronously to SSPCLK. PRESETn must be asserted LOW for a period long enough to reset the slowest block in the on-chip system, and then taken HIGH again. The SPI/SSP requires PRESETn to be asserted LOW for at least one period of PCLK.

In NXP4330D/Q, PRESETn and nSSPRST is controlled by Reset Controller. User can set up SPI/SSP reset signals by CPU.

22.2.2.3 Interrupts

There are 4 maskable interrupts generated in the SPI/SSP. These are combined to produce one interrupt outputs.

- Receive overrun interrupt
- Receive timeout interrupt
- Receive FIFO interrupt (half full)
- Transmit FIFO interrupt (half empty)

22.2.3 Frame format

Each data frame is between 4 and 16 bits long depending on the size of data programmed, and is transmitted starting with the MSB. There are three basic frame types that can be selected:

- Texas Instruments synchronous serial
- Motorola SPI
- National Semiconductor Microwire.

For all three formats, the serial clock (SSPCLKOUT) is held inactive while the SPI/SSP is idle, and transitions at the programmed frequency only during active transmission or reception of data. The idle state of SSPCLKOUT is utilized to provide

a receive timeout indication that occurs when the receive FIFO still contains data after a timeout period.

For Motorola SPI and National Semiconductor Microwire frame formats, the serial frame (SSPFSSOUT) pin is active LOW, and is asserted (pulled down) during the entire transmission of the frame.

For Texas Instruments synchronous serial frame format, the SSPFSSOUT pin is pulsed for one serial clock period starting at its rising edge, prior to the transmission of each frame. For this frame format, both the SPI/SSP and the off-chip slave device drive their output data on the rising edge of SSPCLKOUT, and latch data from the other device on the falling edge.

Unlike the full-duplex transmission of the other two frame formats, the National Semiconductor Microwire format uses a special master-slave messaging technique, which operates at half-duplex. In this mode, when a frame begins, an 8-bit control

message is transmitted to the off-chip slave. During this transmit, no incoming data is received by the SSP. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent,

responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length

anywhere from 13 to 25 bits.

22.2.3.1 Texas Instruments synchronous serial frame format

Figure 22-2 shows the Texas Instruments synchronous serial frame format for a single transmitted frame.

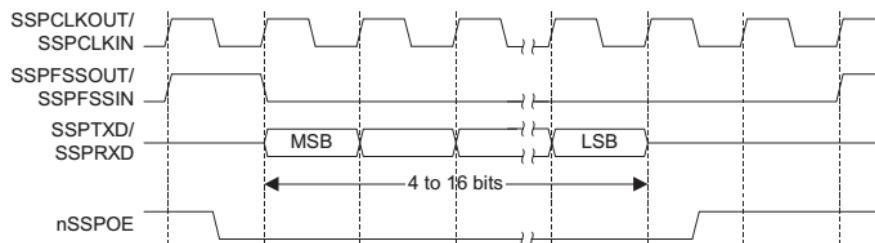


Figure 22-2. Texas Instruments synchronous serial frame format (single transfer)

In this mode, SSPCLKOUT and SSPFSSOUT are forced LOW, and the transmit data line SSPTXD is tristated whenever the SPI/SSP is idle. Once the bottom entry of the transmit FIFO contains data, SSPFSSOUT is pulsed HIGH for one SSPCLKOUT period. The value to be transmitted is also transferred from the transmit FIFO to the serial shift register of the transmit logic. On the next rising edge of SSPCLKOUT, the MSB of the 4 to 16-bit data frame is shifted out on the SSPTXDpin. Likewise, the MSB of the received data is shifted onto the SSPRXDpin by the off-chip serial slave device.

Both the SPI/SSP and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each SSPCLKOUT. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of SSPCLKOUT after the LSB has been latched.

Figure 22-3 shows the Texas Instruments synchronous serial frame format when back-to-back frames are transmitted.

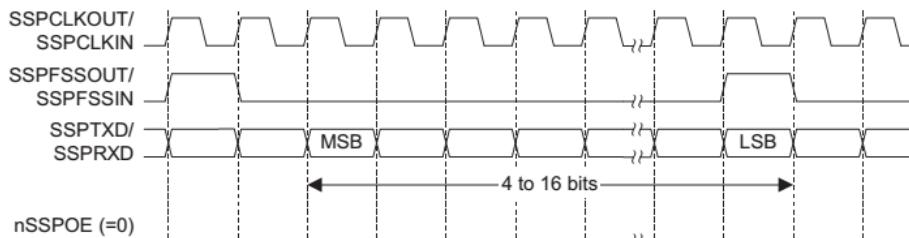


Figure 22-3. TI synchronous serial frame format (continuous transfer)

22.2.3.2 Motorola SPI frame format

The Motorola SPI interface is a four-wire interface where the SSPFSSOUT signal behaves as a slave select. The main feature of the Motorola SPI format is that the inactive state and phase of the SSPCLKOUT signal are programmable through the SPO and SPH bits within the SSPSCR0 control register.

SPO, clock polarity

When the SPO clock polarity control bit is LOW, it produces a steady state low value on the SSPCLKOUT pin. If the SPO clock polarity control bit is HIGH, a steady state high value is placed on the SSPCLKOUT pin when data is not being transferred.

SPH, clock phase

The SPH control bit selects the clock edge that captures data and allows it to change state. It has the most impact on

the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge.

When the SPH phase control bit is LOW, data is captured on the first clock edge transition. If the SPH clock phase control bit is HIGH, data is captured on the second clock edge transition. SSPCR0 register is used to

22.2.3.3 Motorola SPI Format with SPO=0, SPH=0

Single and continuous transmission signal sequences for Motorola SPI format with SPO=0, SPH=0 are shown in Figure 22-4 and Figure 22-5.

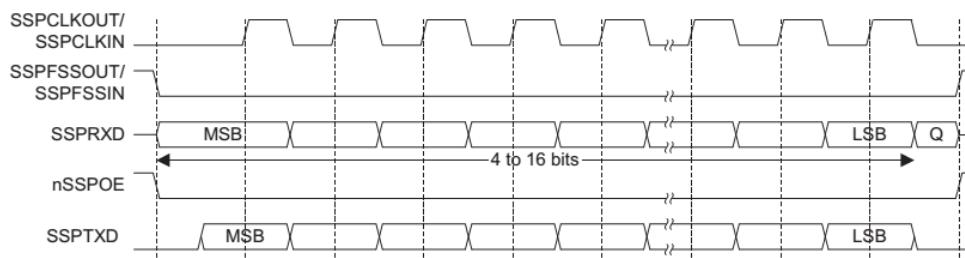


Figure 22-4. Motorola SPI frame format (single transfer) with SPO=0 and SPH=0

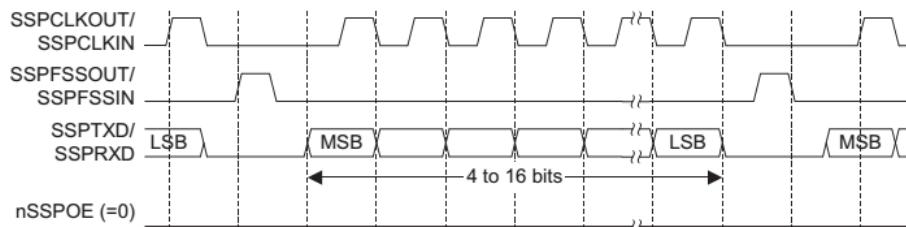


Figure 22-5. Motorola SPI frame format (continuous transfer) with SPO=0 and SPH=0

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. This causes slave data to be enabled onto the SSPRXD input line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

One half SSPCLKOUT period later, valid master data is transferred to the SSPTXD pin. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin goes HIGH after one further half SSPCLKOUT period.

The data is now captured on the rising and propagated on the falling edges of the SSPCLKOUT signal.

In the case of a single word transmission, after all bits of the data word have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the

continuous transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

22.2.3.4 Motorola SPI Format with SPO=0, SPH=1

The transfer signal sequence for Motorola SPI format with SPO=0, SPH=1 is shown in Figure 22-6, which covers both single and continuous transfers.

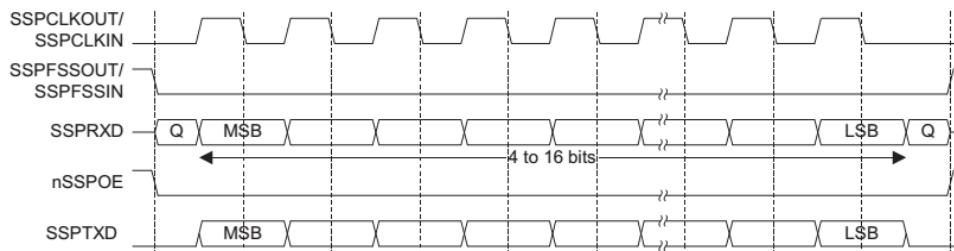


Figure 22-6. Motorola SPI frame format with SPO=0 and SPH=1

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After a further one half SSPCLKOUT period, both master and slave valid data is enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a rising edge transition.

Data is then captured on the falling edges and propagated on the rising edges of the SSPCLKOUT signal.

In the case of a single word transfer, after all bits have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

22.2.3.5 Motorola SPI Format with SPO=1, SPH=0

Single and continuous transmission signal sequences for Motorola SPI format with SPO=1, SPH=0 are shown in Figure 22-7 and Figure 22-8.

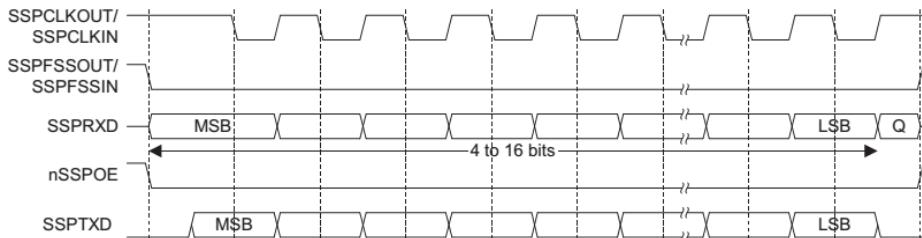


Figure 22-7. Motorola SPI frame format (single transfer) with SPO=1 and SPH=0

Note : In Figure 22-7, Q is an undefined signal

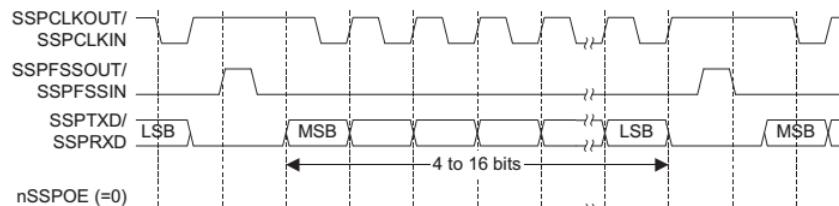


Figure 22-8. Motorola SPI frame format (continuous transfer) with SPO=1 and SPH=0

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced HIGH
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW, which causes slave data to be immediately transferred onto the SSPRXD line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

One half period later, valid master data is transferred to the SSPTXD line. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin becomes LOW after one further half SSPCLKOUT period. This means that data is captured on the falling edges and be propagated on the rising edges of the SSPCLKOUT signal.

In the case of a single word transmission, after all bits of the data word are transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

22.2.3.6 Motorola SPI Format with SPO=1, SPH=1

The transfer signal sequence for Motorola SPI format with SPO=1, SPH=1 is shown in Figure 22-9, which covers both single and continuous transfers.

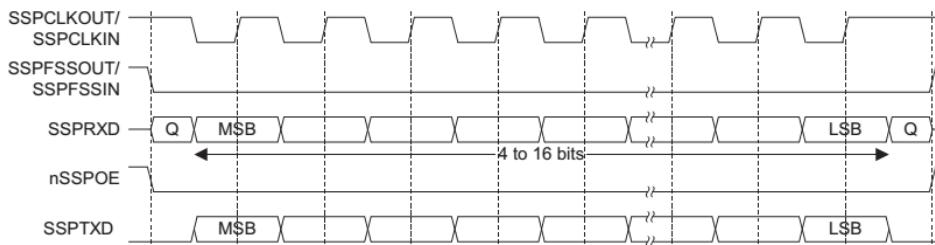


Figure 22-9. Motorola SPI frame format with SPO=1 and SPH=1

Note : In Figure 22-9, Q is an undefined signal

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced HIGH

- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW

If the SPI/SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After a further one half SSPCLKOUT period, both master and slave data are enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a falling edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SSPCLKOUT signal.

After all bits have been transferred, in the case of a single word transmission, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

For continuous back-to-back transmissions, the SSPFSSOUT pins remains in its active LOW state, until the final bit of the last word has been captured, and then returns to its idle state as described above.

For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

22.2.3.7 National Semiconductor Microwire frame format

Figure 22-10 shows the National Semiconductor Microwire frame format, again for a single frame. Figure 22-11 shows the same format when back to back frames are transmitted.

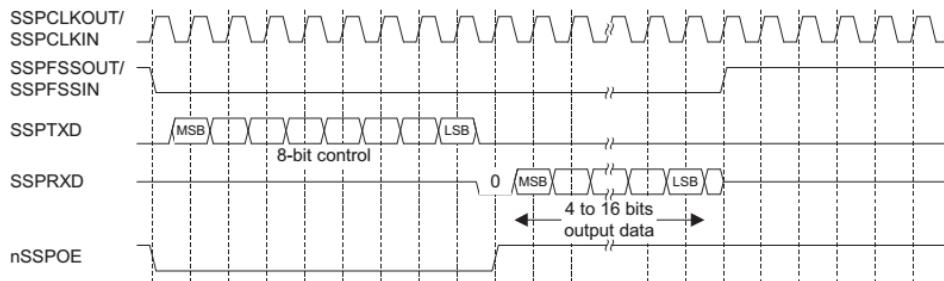


Figure 22-10. Microwire frame format (single transfer)

Microwire format is very similar to SPI format, except that transmission is half-duplex instead of full-duplex, using a master-slave message passing technique. Each serial transmission begins with an 8-bit control word that is transmitted from the SPI/SSP to the off-chip slave device. During this transmission, no incoming data is received by the SPI/SSP. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the required data. The returned data is 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SSPFSSOUT causes the value contained in the bottom entry of the transmit FIFO to be transferred to the serial shift register of the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SSPTXD pin. SSPFSSOUT remains LOW for the

duration of the frame transmission. The SSPRXD pin remains tristated during this transmission.

The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SSPCLKOUT. After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SPI/SSP. Each bit is driven onto SSPRXD line on the falling edge of SSPCLKOUT. The SPI/SSP in turn latches each bit on the rising edge of SSPCLKOUT. At the end of the frame, for single transfers, the SSPFSSOUT signal is pulled HIGH one clock period after the last bit has been latched in the receive serial shifter, that causes the data to be transferred to the receive FIFO.

Note : The off-chip slave device can tristate the receive line either on the falling edge of SSPCLKOUT after the LSB has been latched by the receive shifter, or when the SSPFSSOUT pin goes HIGH.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SSPFSSOUT line is continuously asserted (held LOW) and transmission of data occurs back to back. The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge SSPCLKOUT, after the LSB of the frame has been latched into the SPI/SSP.

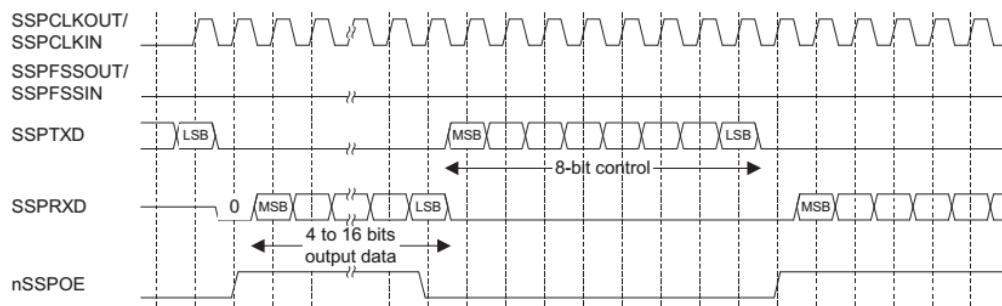


Figure 22-11. Microwire frame format (continuous transfers)

Setup and hold time requirements on SSPFSSIN with respect to SSPCLKIN in Microwire mode

In the Microwire mode, the SPI/SSP slave samples the first bit of receive data on the rising edge of SSPCLKIN after SSPFSSIN has gone LOW. Masters that drive a free-running SSPCLKIN must ensure that the SSPFSSIN signal has sufficient setup and hold margins with respect to the rising edge of SSPCLKIN.

Figure 22-12 illustrates these setup and hold time requirements. With respect to the SSPCLKIN rising edge on which the first bit of receive data is to be sampled by the SPI/SSP slave, SSPFSSIN must have a setup of at least two times the period of SSPCLKIN on which the SPI/SSP operates. With respect to the SSPCLKIN rising edge previous to this edge, SSPFSSIN must have a hold of at least one SSPCLKIN period.

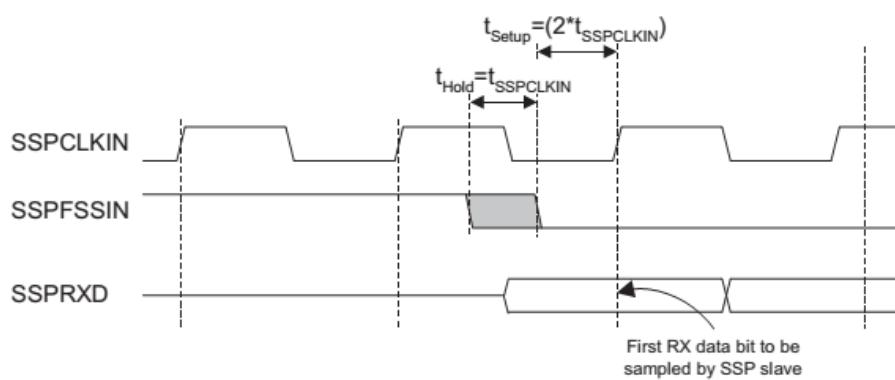


Figure 22-12. Microwire frame format, SSPFSSIN input setup and hold requirements

22.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value														
SPI/SSP Control register 0 (sspcr0)																		
Address : SPISSP0: C005 B000h / SPISSP1 : C005 C000h / SPISSP2 : C005 F000h																		
[31:16]	R	RESERVED	Reserved	16'b0														
[15:8]	R/ W	SCR	<p>Serial clock rate. The value SCR is used to generate the transmit and receive bit rate of the SPI/SSP. The bit rate is</p> $F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$ <p>where CPSDVR is an even value from 2 to 254, programmed through the SSPCPSR register and SCR is a value from 0 to 255.</p>	8'b0														
[7]	R/ W	SPH	SSPCLKOUT phase (applicable to Motorola SPI frame format only)	1'b0														
[6]	R/ W	SPO	SSPCLKOUT polarity (applicable to Motorola SPI frame format only)	1'b0														
[5:4]	R/ W	FRF	<p>Frame format:</p> <p>0 : Motorola SPI frame format 1 : TI synchronous serial frame format 2 : National Microwire frame format 3 : Reserved, undefined operation</p>	1'b0														
[3:0]	R/ W	DSS	<p>Data Size Select</p> <table> <tr> <td>0-2 = Reserved, undefined operation</td> <td>3 : 4-bit data</td> </tr> <tr> <td>4 : 5-bit data</td> <td>5 : 6-bit data</td> <td>6 : 7-bit data</td> <td>7 : 8-bit data</td> </tr> <tr> <td>8 : 9-bit data</td> <td>9 : 10-bit data</td> <td>10 : 11-bit data</td> <td>11 : 12-bit data</td> </tr> <tr> <td>12 : 13-bit data</td> <td>13 : 14-bit data</td> <td>14 : 15-bit data</td> <td>15 : 16-bit data</td> </tr> </table>	0-2 = Reserved, undefined operation	3 : 4-bit data	4 : 5-bit data	5 : 6-bit data	6 : 7-bit data	7 : 8-bit data	8 : 9-bit data	9 : 10-bit data	10 : 11-bit data	11 : 12-bit data	12 : 13-bit data	13 : 14-bit data	14 : 15-bit data	15 : 16-bit data	1'b0
0-2 = Reserved, undefined operation	3 : 4-bit data																	
4 : 5-bit data	5 : 6-bit data	6 : 7-bit data	7 : 8-bit data															
8 : 9-bit data	9 : 10-bit data	10 : 11-bit data	11 : 12-bit data															
12 : 13-bit data	13 : 14-bit data	14 : 15-bit data	15 : 16-bit data															
SPI/SSP Control register 1 (sspcr1)																		
Address : SPISSP0: C005 B004h / SPISSP1 : C005 C004h / SPISSP2 : C005 F004h																		
[31:4]	R	RESERVED	Reserved	28'b0														
[3]	R/ W	SOD	<p>Slave-mode output disable. This bit is relevant only in the slave mode (MS=1). In multiple-slave systems, it is possible for an SPI/SSP master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems the RXD lines from multiple slaves could be tied together. To operate in such systems, the SOD bit can be set if the SPI/SSP slave is not supposed to drive the SSPTXD line.</p> <p>0 : SSP can drive the SSPTXD output in slave mode. 1 : SSP must not drive the SSPTXD output in slave mode.</p>	1'b0														
[2]	R/ W	MS	<p>Master or slave mode select. This bit can be modified only when the SPI/SSP is disabled (SSE=0):</p> <p>0 : device configured as master (default) 1 : device configured as slave.</p>	1'b0														
[1]	R/ W	SSE	<p>Synchronous serial port enable:</p> <p>0 : SSP operation disabled 1 : SSP operation enabled.</p>	1'b0														
[0]	R/ W	LBM	<p>Loop back mode:</p> <p>0 : Normal serial port operation enabled 1 : Output of transmit serial shifter is connected to input of receive serial shifter internally.</p>	1'b0														

Bit	R/W	Symbol	Description	Reset Value
SPI/SSP data register (sspdr)				
Address : SPISSP0: C005 B008h / SPISSP1 : C005 C008h / SPISSP2 : C005 F008h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	DATA	<p>Transmit/Receive FIFO: Read : Receive FIFO Write : Transmit FIFO.</p> <p>You must right-justify data when the SPI/SSP is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by transmit logic. The receive logic automatically right-justifies.</p>	
SPI/SSP STATUS REGISTER (SSPSR)				
Address : SPISSP0: C005 B00Ch / SPISSP1 : C005 C00Ch / SPISSP2 : C005 F00Ch				
[31:5]	R	RESERVED	Reserved	27'b0
[4]	R	BSY	<p>SPI/SSP busy flag (read-only): 0: SSP is idle 1 : SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.</p>	1'b0
[3]	R	RFF	<p>Receive FIFO full (read-only): 0: Receive FIFO is not full 1 : Receive FIFO is full.</p>	1'b0
[2]	R	RNE	<p>Receive FIFO not empty (read-only): 0: Receive FIFO is empty 1 : Receive FIFO is not empty.</p>	1'b0
[1]	R	TNF	<p>Transmit FIFO not full (read-only): 0: Transmit FIFO is full 1 : Transmit FIFO is not full.</p>	1'b1
[0]	R	TFE	<p>Transmit FIFO empty (read-only): 0: Transmit FIFO is not empty 1 : Transmit FIFO is empty.</p>	1'b1
SPI/SSP clock prescaler register (sspcpsr)				
Address : SPISSP0: C005 B010h / SPISSP1 : C005 C010h / SPISSP2 : C005 F010h				
[31:8]	R	RESERVED	Reserved	24'b0
[7:0]	R/W	CPSDVSR	Clock prescale divisor. Must be an even number from 2 to 254, depending on the frequency of SSPCLK. The least significant bit always returns zero on reads	8'b0
SPI/SSP interrupt mask set or clear register (sspirmsc)				
Address : SPISSP0: C005 B014h / SPISSP1 : C005 C014h / SPISSP2 : C005 F014h				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	TXIM	<p>Transmit FIFO interrupt mask: 0 : Tx FIFO half empty or less condition interrupt is masked 1 : Tx FIFO half empty or less condition interrupt is not masked.</p>	1'b0
[2]	R/W	RXIM	<p>Receive FIFO interrupt mask: 0 : Rx FIFO half full or less condition interrupt is masked 1 : Rx FIFO half full or less condition interrupt is not masked.</p>	1'b0
[1]	R/W	RTIM	<p>Receive timeout interrupt mask: 0 : RxFIFO not empty and no read prior to timeout period interrupt is masked 1: RxFIFO not empty and no read prior to timeout period interrupt is not masked.</p>	1'b0
[0]	R/W	RORIM	<p>Receive overrun interrupt mask: 0: RxFIFO written to while full condition interrupt is masked 1: RxFIFO written to while full condition interrupt is not masked.</p>	1'b0
SPI/SSP RAW INTERRUPT STATUS REGISTER (SSPRIS)				

Bit	R/W	Symbol	Description	Reset Value
Address : SPISSP0: C005 B018h / SPISSP1 : C005 C018h / SPISSP2 : C005 F018h				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R	TXRIS	Gives the raw interrupt state (prior to masking) of the SSPTXINTR interrupt	1'b0
[2]	R	RXRIS	Gives the raw interrupt state (prior to masking) of the SSPRXINTR interrupt	1'b0
[1]	R	RTRIS	Gives the raw interrupt state (prior to masking) of the SSPRTINTR interrupt	1'b0
[0]	R	RORRIS	Gives the raw interrupt state (prior to masking) of the SSPRORINTR interrupt	1'b0
SPI/SSP MASKED INTERRUPT STATUS REGISTER (SSPMIS)				
Address : SPISSP0: C005 B01Ch / SPISSP1 : C005 C01Ch / SPISSP2 : C005 F01Ch				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R	TXMIS	Gives the transmit FIFO masked interrupt state (after masking) of the SSPTXINTR Interrupt	1'b0
[2]	R	RXMIS	Gives the receive FIFO masked interrupt state (after masking) of the SSPRXINTR interrupt	1'b0
[1]	R	RTMIS	Gives the receive timeout masked interrupt state (after masking) of the SSPRTINTR Interrupt	1'b0
[0]	R	RORMIS	Gives the receive over run masked interrupt status (after masking) of the SSPRORINTR Interrupt	1'b0
SPI/SSP interrupt clear register (sspicr)				
Address : SPISSP0: C005 B020h / SPISSP1 : C005 C020h / SPISSP2 : C005 F020h				
[31:2]	R	RESERVED	Reserved	30'b0
[1]	W	RTIC	Clears the SSPRTINTR interrupt	
[0]	W	RORIC	Clears the SSPRORINTR interrupt	
SPI/SSP DMA CONTROL REGISTER (SSPDMACR)				
Address : SPISSP0: C005 B024h / SPISSP1 : C005 C024h / SPISSP2 : C005 F024h				
[31:2]	R	RESERVED	Reserved	30'b0
[1]	R/ W	TXDMAE	If this bit is set to 1, DMA for the transmit FIFO is enabled	1'b0
[0]	R/ W	RXDMAE	If this bit is set to 1, DMA for the receive FIFO is enabled	1'b0

Section 23. MPEG-TS Interface

23.1 Overview

The MPEG I/F block receives the output of the MPEG transport decoder chip and then Store the transmitted data the Main Memory using the Fast-DMA of the NXP4330D/Q.

23.1.1 Features

- Supports 1bit / 8bit Modes
- Supports External / Internal DMA
- Supports AES / CAS Encoding & Decoding
- Supports 2 ch MPEG TS interface input
- Supports 1 ch MPEG TS interface output

23.2 Functional Description

23.2.1 Timing

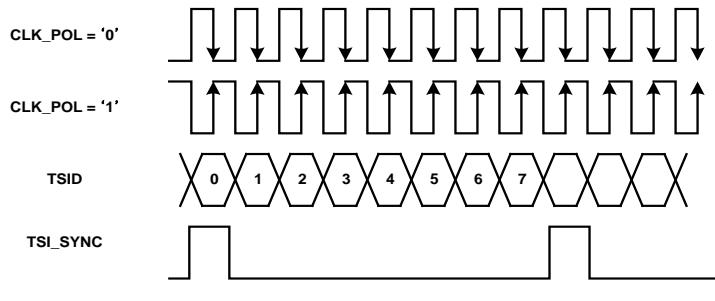


Figure 23-1 Basic Transfer Timing Diagram

Figure 23-1 shows the timing chart in which the NXP4330D/Q reads data when an external device sends data via the MPEG TS interface. If the external device sends data on the rising edge, the NXP4330D/Q actually reads the data on the falling edge

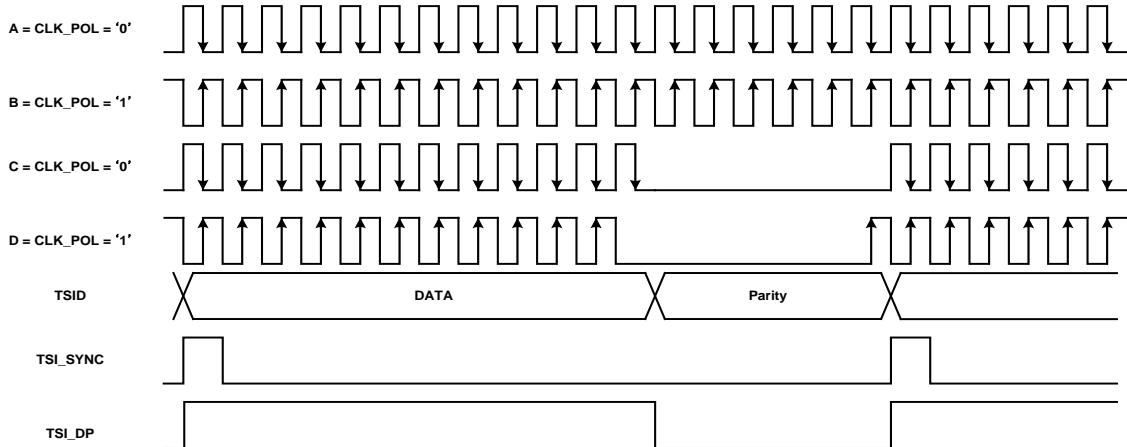


Figure 23-2 MPEG TS Timing at Serial Mode

External devices can provide clocks as well as MPEG TS timing. Figure 23-2 shows the types of clock that can be provided by external devices. The clock types are determined by the point at which the external devices output data.

- A type devices output data on the rising edge. The value of the CAP_CTRL.CAP_CLK_POL bit is '0', and the MPEG TSP reads data on the point of the falling edge as above described.
- B type devices output data on the falling edge. The value of the CAP_CTRL.CAP_CLK_POL bit is '1'.
- C type Devices very similar to the A type device are used. The point of difference from an A type device is that these devices do not check Parity.
- D type Devices very similar to the B type device are used. The point of difference from a B type device is that these devices do not check Parity.

If the value of **TSI_DP** is '1', the value indicates Data. If the value is '0', the value indicates Parity. However, the value may have the opposite meaning depending on device properties. In this case, its polarity can be changed by setting the **MPEGIFCONT.DP_POL** bit as '1'.

23.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
MPEG TSP Capture 0 Control Register 0 (CAP_CTRL0)				
Address : C005_D000h				
[31:28]	-	RESERVED	Reserved	5'b0
[27]	R	CAP0_LOCK_INT_PEND	MPEG TS Interface capture 0 interrupt pending	1'b0
[26]	R/W	CAP0_LOCK_INT_MASK	MPEG TS Interface capture 0 interrupt mask 0 : Interrupt masking enabled 1 : Interrupt masking disabled	1'b0
[25]	R/W	CAP0_LOCK_INT_ENB	MPEG TS Interface capture 0 interrupt enabled 0: Interrupt disabled 1: Interrupt enabled	1'b0
[24]	R/W	CAP0_LOCK_MODE	MPEG TS Interface capture 0 lock mode	1'b0
[23:10]	-	RESERVED	Reserved	24'b0
[9]	R/W	CAP0_SRAM_PWR_ENB	MPEG TS Interface capture 0 SRAM power enable 0 : Power disabled 1 : Power enabled	1'b0
[8]	R/W	CAP0_SRAM_SLEEP	MPEG TS Interface capture 0 SRAM sleep mode 0: Sleep 1: Awake	1'b0
[7]	R/W	CAP0_ERR_POL	MPEG TS Interface capture 0 ERR polling mode 0: negedge captuer 1: posedge capture	1'b0
[6]	R/W	CAP0_SYNC_POL	MPEG TS Interface capture 0 SYNC polling mode 0: negedge captuer 1: posedge capture	1'b0
[5]	R/W	CAP0_DP_POL	MPEG TS Interface capture 0 DP polling mode 0: negedge captuer 1: posedge capture	1'b0
[4]	R/W	CAP0_CLK_POL	MPEG TS Interface capture 0 CLK polling mode 0: negedge captuer 1: posedge capture	1'b0
[3:2]	-	RESERVED	Reserved	2'b0
[1]	R/W	CAP0_CAP_MODE	MPEG TS Interface capture 0 capture mode 0: Byte mode 1: Serial Mode	1'b0
[0]	R/W	CAP0_CAP_ENB	MPEG TS Interface capture 0 capture enable 0: Disable 1: Enable	1'b0
MPEG TSP Capture 1 Control Register 0 (CAP_CTRL1)				
Address : C005_D004h				
[31:28]	-	RESERVED	Reserved	5'b0
[27]	R	CAP1_LOCK_INT_PEND	MPEG TS Interface capture 1 interrupt pending	1'b0
[26]	R/W	CAP1_LOCK_INT_MASK	MPEG TS Interface capture 1 interrupt mask 0 : Interrupt masking enabled 1 : Interrupt masking disabled	1'b0
[25]	R/W	CAP1_LOCK_INT_ENB	MPEG TS Interface capture 1 interrupt enabled 0: Interrupt disabled 1: Interrupt enabled	1'b0
[24]	R/W	CAP1_LOCK_MODE	MPEG TS Interface capture 1 lock mode	1'b0
[23:10]	-	RESERVED	Reserved	24'b0

Bit	R/W	Symbol	Description	Reset Value
[9]	R/W	CAP1_SRAM_PWR	MPEG TS Interface capture 1 SRAM power enable 0: Power disabled 1 : Power enabled	1'b0
[8]	R/W	CAP1_SRAM_SLEEP	MPEG TS Interface capture 1 SRAM sleep mode 0: Sleep 1: Awake	1'b0
[7]	R/W	CAP1_ERR_POL	MPEG TS Interface capture 1 ERR polling mode 0: negedge captuer 1: posedge capture	1'b0
[6]	R/W	CAP1_SYNC_POL	MPEG TS Interface capture 1 SYNC polling mode 0: negedge captuer 1: posedge capture	1'b0
[5]	R/W	CAP1_DP_POL	MPEG TS Interface capture 1 DP polling mode 0: negedge captuer 1: posedge capture	1'b0
[4]	R/W	CAP1_CLK_POL	MPEG TS Interface capture 1 CLK polling mode 0: negedge captuer 1: posedge capture	1'b0
[3:2]	-	RESERVED	Reserved	2'b0
[1]	R/W	CAP1_CAP_MODE	MPEG TS Interface capture 1 capture mode 0: Byte mode 1: Serial Mode	1'b0
[0]	R/W	CAP1_CAP_ENB	MPEG TS Interface capture 1 capture enable 0: Disable 1: Enable	1'b0
MPEG TSP Write PID Value Register (CAP_WR_PID_VAL)				
Address : C005_D008h				
[31:0]	R/W	PID_VALUE	MPEG TSP PID value	32b0
MPEG TSP Write PID Address Register (CAP_WR_PID_ADDR)				
Address : C005_D00Ch				
[31:11]	-	RESERVED	Reserved	21'b0
[10:9]	R/W	PID_WR_SEL	PID write module select 0 : MPEG TSP Capture 0 1 : MPEG TSP Capture 1 2 : MPEG TSP Core module	2'b0
[8:0]	R/W	PID_WR_ADDR	PID Write Address	9'b0
MPEG TSP Capture 0 Caputre Data (CAP0_CAPDATA)				
Address : C005_D010h				
[31:0]	R	CAP0_CAPDATA	MPEG TS Interface capture data	32b0
MPEG TSP Capture 1 Caputre Data (CAP1_CAPDATA)				
Address : C005_D014h				
[31:0]	R	CAP1_CAPDATA	MPEG TS Interface capture data	32b0
MPEG TSP Transfer Data (CORE_TRDATA)				
Address : C005_D01Ch				
[31:0]	R	CORE_TRDATA	MPEG TS Interface transfer data	32b0
MPEG TSP Core Control Register (CORE_CTRL)				
Address : C005_D020h				
[31:19]	-	RESERVED	Reserved	13'b0

Bit	R/W	Symbol	Description	Reset Value
[18]	R	CORE_INT_PEND	MPEG TS Core interrupt pending	1'b0
[17]	R/W	CORE_INT_MASK	MPEG TS Core interrupt mask	1'b0
[16]	R/W	CORE_INT_ENB	MPEG TS Core interrupt enable	1'b0
[15:8]	-	RESERVED	Reserved	8'b0
[7]	R/W	CORE_SRAM_PWR	MPEG TS Core SRAM power	1'b0
[6]	R/W	CORE_SRAM_SLEEP	MPEG TS Core SRAM sleep mode	1'b0
[5:2]	-	RESERVED	Reserved	4'b0
[1]	R/W	CORE_ENCR_MODE	MPEG TS Core Encryption mode 0: Decoding 1: Encoding	1'b0
[0]	R/W	CORE_ENB	MPEG TS Core Enable	1'b0
MPEG TSP IDMA Status Register (IDMA_STATUS)				
Address : C005_D024h				
[31:20]	-	RESERVED	Reserved	12'b0
[19]	R	IDMA3_BUSY	Internal DMA3 busy	1'b0
[18]	R	IDMA2_BUSY	Internal DMA2 busy	1'b0
[17]	R	IDMA1_BUSY	Internal DMA1 busy	1'b0
[16]	R	IDMA0_BUSY	Internal DMA0 busy	1'b0
[15:4]	-	RESERVED	Reserved	12'b0
[3]	R/W	IDMA3_ENB	Internal DMA3 Enable	1'b0
[2]	R/W	IDMA2_ENB	Internal DMA2 Enable	1'b0
[1]	R/W	IDMA1_ENB	Internal DMA1 Enable	1'b0
[0]	R/W	IDMA0_ENB	Internal DMA0 Enable	1'b0
MPEG TSP IDMA Control Register (IDMA_CON)				
Address : C005_D028h				
[31:20]	-	RESERVED	Reserved	12'b0
[19]	W	IDMA3_STOP	Internal DMA3 stop	1'b0
[18]	W	IDMA2_STOP	Internal DMA2 stop	1'b0
[17]	W	IDMA1_STOP	Internal DMA1 stop	1'b0
[16]	W	IDMA0_STOP	Internal DMA0 stop	1'b0
[15:4]	-	RESERVED	Reserved	12'b0
[3]	W	IDMA3_RUN	Internal DMA3 run	1'b0
[2]	W	IDMA2_RUN	Internal DMA2 run	1'b0
[1]	W	IDMA1_RUN	Internal DMA1 run	1'b0
[0]	W	IDMA0_RUN	Internal DMA0 run	1'b0
MPEG TSP IDMA Interrupt Register (IDMA_INT)				
Address : C005_D02Ch				
[31:28]	-	RESERVED	Reserved	4'b0

Bit	R/W	Symbol	Description	Reset Value
[27]	R/W	IDMA3_INT_ENB	Internal DMA3 interrupt enable	1'b0
[26]	R/W	IDMA2_INT_ENB	Internal DMA2 interrupt enable	1'b0
[25]	R/W	IDMA1_INT_ENB	Internal DMA1 interrupt enable	1'b0
[24]	R/W	IDMA0_INT_ENB	Internal DMA0 interrupt enable	1'b0
[23:20]	-	RESERVED	Reserved	4'b0
[19]	R/W	IDMA3_INT_MASK	Internal DMA3 interrupt mask	1'b0
[18]	R/W	IDMA2_INT_MASK	Internal DMA2 interrupt mask	1'b0
[17]	R/W	IDMA2_INT_MASK	Internal DMA1 interrupt mask	1'b0
[16]	R/W	IDMA0_INT_MASK	Internal DMA0 interrupt mask	1'b0
[15:12]	-	RESERVED	Reserved	4'b0
[11]	R/W	IDMA3_INT_PEND	Internal DMA3 interrupt pending	1'b0
[10]	R/W	IDMA2_INT_PEND	Internal DMA2 interrupt pending	1'b0
[9]	R/W	IDMA2_INT_PEND	Internal DMA1 interrupt pending	1'b0
[8]	R/W	IDMA0_INT_PEND	Internal DMA0 interrupt pending	1'b0
[7:4]	-	RESERVED	Reserved	4'b0
[3]	W	IDMA3_INT_CLR	Internal DMA3 interrupt pending clear	1'b0
[2]	W	IDMA2_INT_CLR	Internal DMA2 interrupt pending clear	1'b0
[1]	W	IDMA2_INT_CLR	Internal DMA1 interrupt pending clear	1'b0
[0]	W	IDMA0_INT_CLR	Internal DMA0 interrupt pending clear	1'b0

MPEG TSP Internal DMA0 Base Address Register (IDMA0_ADDR)

Address : C005_D030

[31:0]	R/W	IDMA0_ADDR	Internal DMA0 Base Addr	32b0
--------	-----	------------	-------------------------	------

MPEG TSP Internal DMA1 Base Address Register (IDMA1_ADDR)

Address : C005_D034

[31:0]	R/W	IDMA1_ADDR	Internal DMA1 Base Addr	32b0
--------	-----	------------	-------------------------	------

MPEG TSP Internal DMA2 Base Address Register (IDMA2_ADDR)

Address : C005_D038

[31:0]	R/W	IDMA2_ADDR	Internal DMA2 Base Addr	32b0
--------	-----	------------	-------------------------	------

MPEG TSP Internal DMA3 Base Address Register (IDMA3_ADDR)

Address : C005_D03C

[31:0]	R/W	IDMA3_ADDR	Internal DMA3 Base Addr	32b0
--------	-----	------------	-------------------------	------

MPEG TSP Internal DMA0 Data Length Register (IDMA0_LEN)

Address : C005_D040

[31:0]	R/W	IDMA0_LEN	Internal DMA0 Data Length	32b0
--------	-----	-----------	---------------------------	------

MPEG TSP Internal DMA1 Data Length Register (IDMA1_LEN)

Address : C005_D044

[31:0]	R/W	IDMA1_LEN	Internal DMA1 Data Length	32b0
--------	-----	-----------	---------------------------	------

Bit	R/W	Symbol	Description	Reset Value
MPEG TSP Internal DMA2 Data Length Register (IDMA2_LEN)				
Address : C005_D048				
[31:0]	R/W	IDMA2_LEN	Internal DMA2 Data Length	32b0
MPEG TSP Internal DMA3 Data Length Register (IDMA3_LEN)				
Address : C005_D04C				
[31:0]	R/W	IDMA3_LEN	Internal DMA3 Data Length	32b0

Section 24. **UART_ISO7816**

24.1 Overview

The UART performs:

- serial-to-parallel conversion on data received from a peripheral device
- parallel-to-serial conversion on data transmitted to the peripheral device.

The CPU reads and writes data and control/status information through the AMBA APB interface. The transmit and receive paths are buffered with internal FIFO memories enabling up to 32-bytes to be stored independently in both transmit and receive modes.

The UART:

- includes a programmable baud rate generator that generates a common transmit and receive internal clock from the UART internal reference clock input, UARTCLK
- offers similar functionality to the industry-standard 16C650 UART device
- supports the following maximum baud rates:
 - 921600 bps, in UART mode
 - 460800 bps, in IrDA mode
 - 115200 bps, in low-power IrDA mode.

The UART operation and baud rate values are controlled by the Line Control Register, UARTLCR_H and the baud rate divisor registers (Integer Baud Rate Register, UARTIBRD and Fractional Baud Rate Register, UARTRFRD).

The UART can generate:

- individually-maskable interrupts from the receive (including timeout), transmit, modem status and error conditions
- a single combined interrupt so that the output is asserted if any of the individual interrupts are asserted, and unmasked
- DMA request signals for interfacing with a Direct Memory Access (DMA)controller.

If a framing, parity, or break error occurs during reception, the appropriate error bit is set, and is stored in the FIFO. If an overrun condition occurs, the overrun register bit is set immediately and FIFO data is prevented from being overwritten.

You can program the FIFOs to be 1-byte deep providing a conventional double-buffered UART interface.

The modem status input signals Clear To Send (CTS), Data Carrier Detect (DCD),Data Set Ready (DSR), and Ring Indicator (RI) are supported. The output modem control lines, Request To Send (RTS), and Data Terminal Ready (DTR) are also supported.

There is a programmable hardware flow control feature that uses the nUARTCTS input and the nUARTRTS output to automatically control the serial data flow.

The device is a cyclic type monolithic ADC, which provides an on-chip sample-and-hold and power down mode.

IrDA SIR block

The IrDA Serial InfraRed (SIR) block contains an IrDA SIR protocol ENDEC. The SIR protocol ENDEC can be enabled for serial communication through signals nSIROUT and SIRIN to an infrared transducer instead of using the UART signals UARTRXD and UARTRXD.

If the SIR protocol ENDEC is enabled, the UARTRXD line is held in the passive state(HIGH) and transitions of the modem status, or the UARTRXD line have no effect. The SIR protocol ENDEC can receive and transmit, but it is half-duplex only, so it cannot receive while transmitting, or transmit while receiving.

The IrDA SIR physical layer specifies a minimum 10ms delay between transmission and reception.

24.1.1 Features

The UART provides:

- Compliance to the AMBA Specification (Rev 2.0) onwards for easy integration into SoC implementation.
- Programmable use of UART or IrDA SIR input/output.
- Separate 32×8 transmit and 32×12 receive First-In, First-Out (FIFO) memory buffers to reduce CPU interrupts.
- Programmable FIFO disabling for 1-byte depth.
- Programmable baud rate generator. This enables division of the reference clock by (1×16) to (65535×16) and generates an internal ×16 clock. The divisor can be a fractional number enabling you to use any clock with a frequency >3.6864MHz as the reference clock.
- Standard asynchronous communication bits (start, stop and parity). These are added prior to transmission and removed on reception.
- Independent masking of transmit FIFO, receive FIFO, receive timeout, modem status, and error condition interrupts.
- Support for Direct Memory Access (DMA) : UART0, UART1, and UART2 only
- False start bit detection.
- Line break generation and detection.
- Support of the modem control functions CTS, DCD, DSR, RTS, DTR, and RI : UART1 only
- Programmable hardware flow control.
- Fully-programmable serial interface characteristics:
 - data can be 5, 6, 7, or 8 bits
 - even, odd, stick, or no-parity bit generation and detection
 - 1 or 2 stop bit generation
 - baud rate generation, dc up to UARTCLK/16
- IrDA SIR ENDEC block providing:
 - programmable use of IrDA SIR or UART input/output
 - support of IrDA SIR ENDEC functions for data rates up to 115200 bps half-duplex
 - support of normal 3/16 and low-power (1.41-2.23μs) bit durations

- programmable division of the UARTCLK reference clock to generate the appropriate bit duration for low-power IrDA mode.
- Support of the ISO-7816.
- Identification registers that uniquely identify the UART. These can be used by an operating system to automatically configure itself.

24.1.2 UART Block Diagram

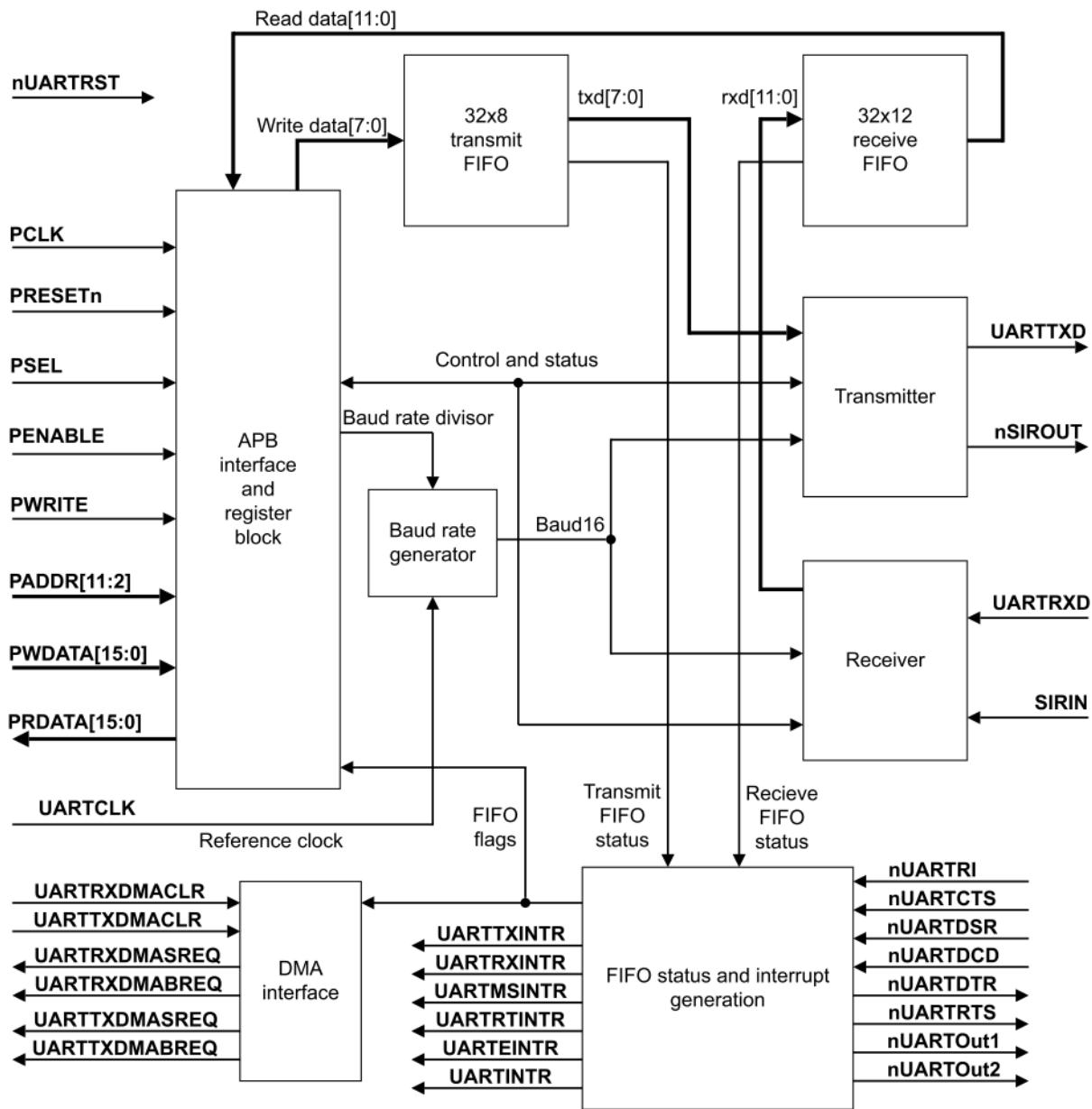


Figure 24-1. UART Block Diagram

AMBA APB interface

The AMBA APB interface generates read and write decodes for accesses to status/control registers, and the transmit and receive FIFOs.

Register block

The register block stores data written, or to be read across the AMBA APB interface.

Baud rate generator

The baud rate generator contains free-running counters that generate the internal $\times 16$ clocks, Baud16 and IrLPBaud16

signals. Baud16 provides timing information for UART transmit and receive control. Baud16 is a stream of pulses with a width of one UARTCLK clock period and a frequency of 16 times the baud rate. IrLPBaud16 provides timing information to generate the pulse width of the IrDA encoded transmit bit stream when in low-power IrDA mode.

Transmit FIFO

The transmit FIFO is an 8-bit wide, 32 location deep, FIFO memory buffer. CPU data written across the APB interface is stored in the FIFO until read out by the transmit logic. You can disable the transmit FIFO to act like a one-byte holding register. The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line break detection are also performed, and their status accompanies the data that is written to the receive FIFO.

Receive FIFO

The receive FIFO is a 12-bit wide, 32 location deep, FIFO memory buffer. Received data and corresponding error bits, are stored in the receive FIFO by the receive logic until read out by the CPU across the APB interface. The receive FIFO can be disabled to act like a one-byte holding register.

Transmit logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. Control logic outputs the serial bit stream beginning with a start bit, data bits with the Least Significant Bit (LSB) first, followed by the parity bit, and then the stop bits according to the programmed configuration in control registers.

Receive logic

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line break detection are also performed, and their status accompanies the data that is written to the receive FIFO.

Interrupt generation logic

Individual maskable active HIGH interrupts are generated by the UART. A combined interrupt output is also generated as an OR function of the individual interrupt requests. You can use the single combined interrupt with a system interrupt controller that provides another level of masking on a per-peripheral basis. This enables you to use modular device drivers that always know where to find the interrupt source control register bits.

You can also use the individual interrupt requests with a system interrupt controller that provides masking for the outputs of each peripheral. In this way, a global interrupt service routine can read the entire set of sources from one wide register in the system interrupt controller. This is attractive where the time to read from the peripheral registers is significant compared to the CPU clock speed in a real-time system.

DMA interface

The UART provides an interface to connect to the DMA controller.

Synchronizing registers and logic

The UART supports both asynchronous and synchronous operation of the clocks, PCLK and UARTCLK. Synchronization registers and handshaking logic have been implemented, and are active at all times. This has a minimal impact on performance or area. Synchronization of control signals is performed on both directions of data flow, that is from the PCLK to the UARTCLK domain, and from the UARTCLK to the PCLK domain.

24.1.3 IrDA SIR ENDEC functional description

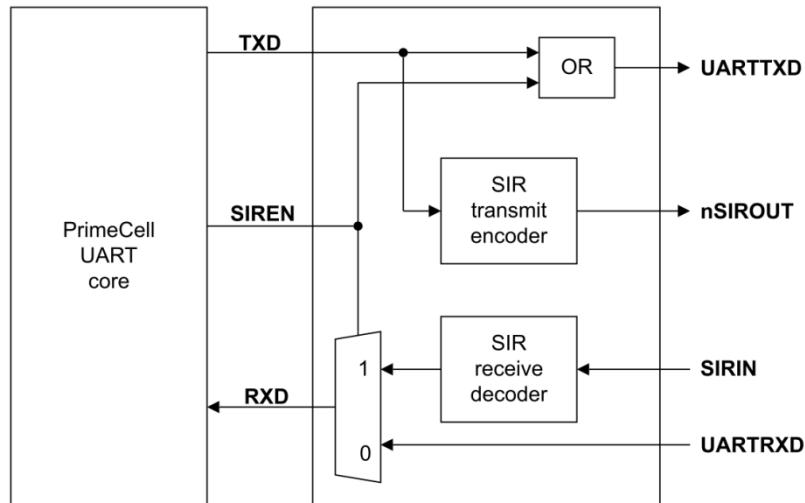


Figure 24-2. IrDA SIR RNDEC Block Diagram

24.1.3.1 IrDA SIR transmit encoder

The SIR transmit encoder modulates the Non Return-to-Zero (NRZ) transmit bit stream output from the UART. The IrDA SIR physical layer specifies use of a Return To Zero, Inverted (RZI) modulation scheme that represents logic 0 as an infrared light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared Light Emitting Diode (LED).

In IrDA mode the transmitted pulse width is specified as three times the period of the internal $\times 16$ clock (Baud16), that is, 3/16 of a bit period.

In low-power IrDA mode the transmit pulse width is specified as 3/16 of a 115200 bps bit period. This is implemented as three times the period of a nominal 1.8432MHz clock(IrLPBaud16) derived from dividing down of UARTCLK clock. The frequency of IrLPBaud16 is set up by writing the appropriate divisor value to the IrDA Low-Power Counter Register, UARTILPR.

The active low encoder output is normally LOW for the marking state (no light pulse).The encoder outputs a high pulse to generate an infrared light pulse representing a logic0 or spacing state.

In normal and low-power IrDA modes, when the fractional baud rate divider is used, the transmitted SIR pulse stream includes an increased amount of jitter. This jitter is because the Baud16 pulses cannot be generated at regular intervals when fractional division is used. That is, the Baud16 cycles have a different number of UARTCLK cycles. It can be shown that the worst case jitter in the SIR pulse stream can be up to three UARTCLK cycles. This is within the limits of the SIR IrDA Specification where the maximum amount of jitter permitted is 13%, provided the UARTCLK is $> 3.6864\text{MHz}$ and the maximum baud rate used for IrDA mode is ≤ 115200 bps. With these conditions, the jitter is less than 9%.

24.1.3.2 IrDA SIR receive decoder

The SIR receive decoder demodulates the return-to-zero bit stream from the infrared detector and outputs the received NRZ serial bit stream to the UART received data input. The decoder input is normally HIGH (marking state) in the idle state. The transmit encoder output has the opposite polarity to the decoder input.

A start bit is detected when the decoder input is LOW.

<Note>

To prevent the UART from responding to glitches on the received data input then it ignores SIRIN pulses that are less than:

- 3/16 of Baud16, in IrDA mode
- 3/16 of IrLPBaud16, in low-power IrDA mode.

24.2 Operation

24.2.1 Interface reset

The UART and IrDA SIR ENDEC are reset by the global reset signal PRESETn and a block-specific reset signal nUARTRST. An external reset controller must use PRESETn to assert nUARTRST asynchronously and negate it synchronously to UARTCLK. PRESETn must be asserted LOW for a period long enough to reset the slowest block in the on-chip system, and then be taken HIGH again. The UART requires PRESETn to be asserted LOW for at least one period of PCLK.

24.2.2 Clock signals

The frequency selected for UARTCLK must accommodate the required range of baudrates:

- FUARTCLK (min) $\geq 16 \times \text{baud_rate(max)}$
- FUARTCLK(max) $\leq 16 \times 65535 \times \text{baud_rate(min)}$

For example, for a range of baud rates from 110 baud to 460800 baud the UARTCLK frequency must be between 7.3728MHz to 115.34MHz.

The frequency of UARTCLK must also be within the required error limits for all baudrates to be used.

There is also a constraint on the ratio of clock frequencies for PCLK to UARTCLK. The frequency of UARTCLK must be no more than 5/3 times faster than the frequency of PCLK:

- FUARTCLK $\leq 5/3 \times \text{FPCLK}$

For example, in UART mode, to generate 921600 baud when UARTCLK is 14.7456MHz then PCLK must be greater than or equal to 8.85276MHz. This ensures that the UART has sufficient time to write the received data to the receive FIFO.

24.2.3 UART operation

Control data is written to the UART Line Control Register, UARTLCR. This register is 30-bits wide internally, but is externally accessed through the APB interface by writes to the following registers:

UARTLCR_H Defines the:

- transmission parameters
- word length
- buffer mode
- number of transmitted stop bits
- parity mode
- break generation.

UARTIBRD Defines the integer baud rate divider

UARTFBRD Defines the fractional baud rate divider

Fractional baud rate divider

The baud rate divisor is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part. This is used by the baud

rate generator to determine the bit period. The fractional baud rate divider enables the use of any clock with a frequency $>3.6864\text{MHz}$ to act as UARTCLK, while it is still possible to generate all the standard baud rates.

The 16-bit integer is written to the Integer Baud Rate Register, UARTIBRD. The 6-bit fractional part is written to the Fractional Baud Rate Register, UARTRFBRD. The Baud Rate Divisor has the following relationship to UARTCLK:

- Baud Rate Divisor = $\text{UARTCLK}/(16 \times \text{Baud Rate}) = \text{BRDI} + \text{BRDF}$

where BRDI is the integer part and BRDF is the fractional part separated by a decimal point as Figure 24-3 shows.

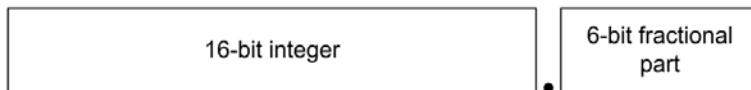


Figure 24-3. Baud rate divisor

You can calculate the 6-bit number (m) by taking the fractional part of the required baudrate divisor and multiplying it by 64 (that is, 2^n , where n is the width of the UARTRFBRD Register) and adding 0.5 to account for rounding errors:

- $m = \text{integer}(\text{BRDF} \times 2^n + 0.5)$

An internal clock enable signal, Baud16, is generated, and is a stream of one UARTCLK wide pulses with an average frequency of 16 times the required baud rate. This signal is then divided by 16 to give the transmit clock. A low number in the baudrate divisor gives a short bit period, and a high number in the baud rate divisor gives along bit period.

Data transmission or reception

Data received or transmitted is stored in two 32-byte FIFOs, though the receive FIFO has an extra four bits per character for status information.

For transmission, data is written into the transmit FIFO. If the UART is enabled, it causes a data frame to start transmitting with the parameters indicated in the LineControl Register, UARTLCR_H. Data continues to be transmitted until there is no data left in the transmit FIFO. The BUSY signal goes HIGH as soon as data is written to the transmit FIFO (that is, the FIFO is non-empty) and remains asserted HIGH while data is being transmitted. BUSY is negated only when the transmit FIFO is empty, and the last character has been transmitted from the shift register, including the stop bits. BUSY can be asserted HIGH even though the UART might no longer be enabled.

For each sample of data, three readings are taken and the majority value is kept. In the following paragraphs the middle sampling point is defined, and one sample is taken either side of it.

When the receiver is idle (UARTRXD continuously 1, in the marking state) and a LOW is detected on the data input (a start bit has been received), the receive counter, with the clock enabled by Baud16, begins running and data is sampled on the eighth cycle of that counter in UART mode, or the fourth cycle of the counter in SIR mode to allow for the shorter logic 0 pulses (half way through a bit period).

The start bit is valid if UARTRXD is still LOW on the eighth cycle of Baud16, otherwise a false start bit is detected and it is ignored.

If the start bit was valid, successive data bits are sampled on every 16th cycle of Baud16 (that is, one bit period later) according to the programmed length of the data characters. The parity bit is then checked if parity mode was enabled.

Lastly, a valid stop bit is confirmed if UARTRXD is HIGH, otherwise a framing error has occurred. When a full word is received, the data is stored in the receive FIFO, with any error bits associated with that word.

Error bits

Three error bits are stored in bits [10:8] of the receive FIFO, and are associated with a particular character. There is an additional error that indicates an overrun error and this is stored in bit 11 of the receive FIFO.

Overrun bit

The overrun bit is not associated with the character in the receive FIFO. The overrun error is set when the FIFO is full, and the next character is completely received in the shift register. The data in the shift register is overwritten, but it is not written into the FIFO. When an empty location is available in the receive FIFO, and another character is received, the state of the overrun bit is copied into the receive FIFO along with the received character. The overrun state is then cleared. Table 24-1 lists the bit functions of the receive FIFO.

FIFO bit	Function
11	Overrun indicator
10	Break error
9	Parity error
8	Framing error
7:0	Received data

Table 24-1. Receive FIFO bit functions

Disabling the FIFOs

Additionally, you can disable the FIFOs. In this case, the transmit and receive sides of the UART have 1-byte holding registers (the bottom entry of the FIFOs). The overrun bit is set when a word has been received, and the previous one was not yet read. In this implementation, the FIFOs are not physically disabled, but the flags are manipulated to give the illusion of a 1-byte register. When the FIFOs are disabled, a write to the data register bypasses the holding register unless the transmit shift register is already in use.

System and diagnostic loopback testing

You can perform loopback testing for UART data by setting the Loop Back Enable (LBE) bit to 1 in the Control Register, UARTCR.

Data transmitted on UARTRXD is received on the UARTRXD input.

IrDA SIR operation

The IrDA SIR ENDEC provides functionality that converts between an asynchronous UART data stream, and half-duplex serial SIR interface. No analog processing is performed on-chip. The role of the SIR ENDEC is to provide a digital encoded output, and decoded input to the UART. There are two modes of operation:

- In IrDA mode, a zero logic level is transmitted as a high pulse of 3/16th duration of the selected baud rate bit period on the nSIROUT signal, while logic one levels are transmitted as a static LOW signal. These levels control the driver of an infrared transmitter, sending a pulse of light for each zero. On the reception side, the incoming light pulses energize the photo transistor base of the receiver, pulling its output LOW. This drives the SIRIN signal LOW.
- In low-power IrDA mode, the width of the transmitted infrared pulse is set to three times the period of the internally generated IrLPBaud16 signal (1.63µs, assuming a nominal 1.8432MHz frequency) by setting the SIRLP bit in the Control Register, UARTCR.

In normal and low-power IrDA modes:

- during transmission, the UART data bit is used as the base for encoding
- during reception, the decoded bits are transferred to the UART receive logic.

The IrDA SIR physical layer specifies a half-duplex communication link, with a minimum 10ms delay between transmission and reception. This delay must be generated by software because it is not supported by the UART. The delay is required because the Infrared receiver electronics might become biased, or even saturated from the optical power coupled from the adjacent transmitter LED. This delay is known as latency, or receiver setup time.

The IrLPBaud16 signal is generated by dividing down the UARTCLK signal according to the low-power divisor value written to the IrDA Low-Power Counter Register, UARTEILPR.

The low-power divisor value is calculated as:

- Low-power divisor = $(F_{UARTCLK} / F_{IrLPBaud16})$

where $F_{IrLPBaud16}$ is nominally 1.8432MHz.

The divisor must be chosen so that $1.42\text{MHz} < F_{IrLPBaud16} < 2.12\text{MHz}$.

System and diagnostic loopback testing

It is possible to perform loopback testing for SIR data by:

setting the LBE bit to 1 in the Control Register, UARTCR.

setting the SIRTEST bit to 1 in the Test Control Register, UARTECCR.

Data transmitted on nSIROUT is received on the SIRIN input.

<Note>

This is the only occasion that a test register is accessed during normal operation.

24.2.4 UART character frame

Figure 24-4 shows the UART character frame.

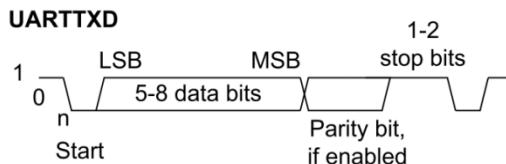


Figure 24-4. UART character frame

24.2.5 IrDA data modulation

Figure 24-5 shows the effect of IrDA 3/16 data modulation.

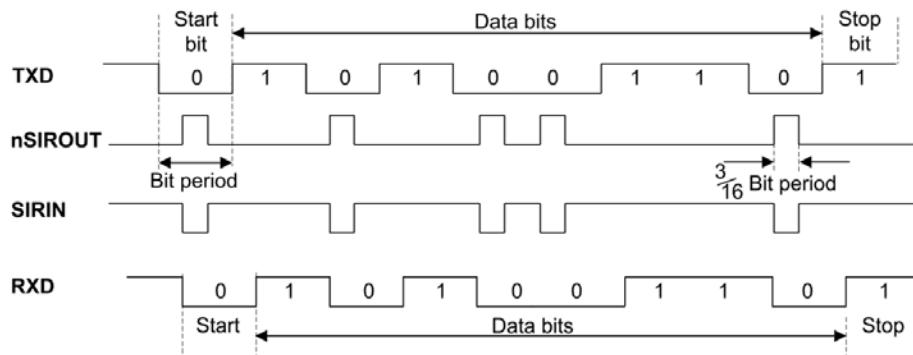


Figure 24-5. IrDA data modulation (3/16)

24.2.6 UART modem operation

You can use the UART to support both the Data Terminal Equipment (DTE) and DataCommunication Equipment (DCE) modes of operation.

Signal	Function	
	DTE	DCE
nUARTCTS	Clear to send	Request to send
nUARTDSR	Data set ready	Data terminal ready
nUARTDCD	Data carrier detect	-
nUARTRI	Ring indicator	-
nUARTRTS	Request to send	Clear to send
nUARTDTR	Data terminal ready	Data set ready
nUARTOUT1	-	Data carrier detect
nUARTOUT2	-	Ring indicator

Table 24-2. Function of the modem input/output signals in DTE and DCE modes

24.2.7 UART DMA interface

The UART provides an interface to connect to a DMA controller. The DMA operationof the UART is controlled using the DMA Control Register, UARTDMACR. The DMA interface includes the following signals:

For receive:

- UARTRXDMASREQ
Single character DMA transfer request, asserted by the UART. Forreceive, one character consists of up to 12 bits. This signal is assertedwhen the receive FIFO contains at least one character.
- UARTRXDMABREQ
Burst DMA transfer request, asserted by the UART. This signal isasserted when the receive FIFO contains more characters than theprogrammed watermark level. You can program the watermark level foreach FIFO using the Interrupt FIFO Level Select Register, UARTIFLS.
- UARTRXDMACLR
DMA request clear, asserted by a DMA controller to clear the receiverrequest signals. If DMA burst transfer is requested, the clear signal isasserted during the transfer of the last data in the burst.

For transmit:

- UARTTXDMASREQ
Single character DMA transfer request, asserted by the UART. Fortransmit one character consists of up to eight bits. This signal is assertedwhen there is at least one empty location in the transmit FIFO.
- UARTTXDMABREQ
Burst DMA transfer request, asserted by the UART. This signal isasserted when the transmit FIFO contains less characters than thewatermark level. You can program the watermark level for each FIFOusing the Interrupt FIFO Level Select Register, UARTIFLS.

■ UARTTXDMACLR

DMA request clear, asserted by a DMA controller to clear the transmitrequest signals. If DMA burst transfer is requested, the clear signal is asserted during the transfer of the last data in the burst.

The burst transfer and single transfer request signals are not mutually exclusive, they can both be asserted at the same time. For example, when there is more data than the watermark level in the receive FIFO, the burst transfer request and the single transferrequest are asserted. When the amount of data left in the receive FIFO is less than the watermark level, the single request only is asserted. This is useful for situations where the number of characters left to be received in the stream is less than a burst.

For example, if 19 characters have to be received and the watermark level is programmed to be four. The DMA controller then transfers four bursts of four characters and three single transfers to complete the stream.

<Note>

For the remaining three characters the UART cannot assert the burst request.

Each request signal remains asserted until the relevant DMACLR signal is asserted. After the request clear signal is deasserted, a request signal can become active again, depending on the conditions described previously. All request signals are deasserted if the UART is disabled or the relevant DMA enable bit, TXDMAE or RXDMAE, in the DMA Control Register, UARTRDMACR is cleared.

If you disable the FIFOs in the UART then it operates in character mode and only the DMA single transfer mode can operate, because only one character can be transferred to, or from the FIFOs at any time. UARTRXDMASREQ and UARTTXDMASREQ are the only request signals that can be asserted. See the Line Control Register, UARTRLCR_H for information about disabling the FIFOs.

When the UART is in the FIFO enabled mode, data transfers can be made by either single or burst transfers depending on the programmed watermark level and the amount of data in the FIFO. Table 24-3 lists the trigger points for UARTRXDMABREQ and UARTTXDMABREQ depending on the watermark level, for the transmit and receive FIFOs.

Watermark level	Burst length	
	Transmit (number of empty locations)	Receive (number of filled locations)
1/8	28	4
1/4	24	8
1/2	16	16
3/4	8	24
7/8	4	28

Table 24-3. DMA trigger points for the transmit and receive FIFOs

In addition, the DMAONERR bit in the DMA Control Register, UARTRDMACR supports the use of the receive error interrupt, UARTEINTR. It enables the DMA receive request outputs, UARTRXDMASREQ or UARTRXDMABREQ, to be masked out when the UART error interrupt, UARTEINTR, is asserted. The DMA receive request outputs remain inactive until the UARTEINTR is cleared. The DMA transmit request outputs are unaffected.

Figure 24-6 shows the timing diagram for both a single transfer request and a burst transfer request with the appropriate DMACLR signal. The signals are all synchronous to PCLK. For the sake of clarity it is assumed that there is no

synchronization of the request signals in the DMA controller.

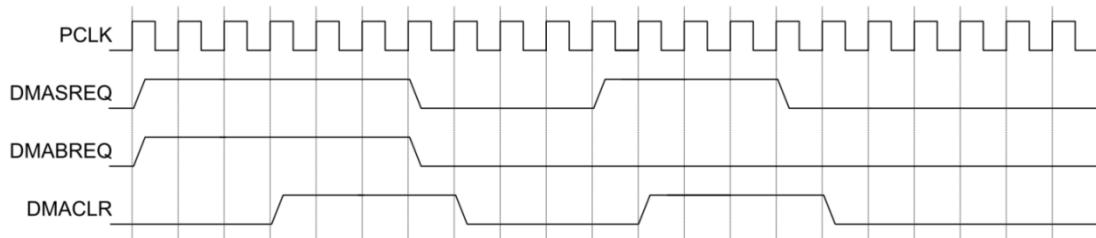


Figure 24-6. DMA transfer waveforms

24.2.8 Interrupts

There are eleven maskable interrupts generated in the UART. These are combined to produce five individual interrupt outputs and one that is the OR of the individual outputs:

- UARTRXINTR
- UARTTXINTR
- UARTRTIINTR
- UARTMSINTR, that can be caused by:
 - UARTRIINTR, because of a change in the nUARTRI modem status
 - UARTCTSINTR, because of a change in the nUARTCTS modem status
 - UARDCCDINTR, because of a change in the nUARTDCD modem status
 - UARDSRINTR, because of a change in the nUARTDSR modem status.
- UARTEINTR, that can be caused by:
 - UARTOEINTR, because of an overrun error
 - UARTBEINTR, because of a break in the reception
 - UARPPEINTR, because of a parity error in the received character
 - UARTFEINTR, because of a framing error in the received character.
- UARTINTR, this is an OR function of the five individual masked outputs.

You can enable or disable the individual interrupts by changing the mask bits in the Interrupt Mask Set/Clear Register, UARTIMSC. Setting the appropriate mask bit HIGH enables the interrupt.

Provision of individual outputs and the combined interrupt output, enables you to use either a global interrupt service routine, or modular device drivers to handle interrupts.

The transmit and receive dataflow interrupts **UARTRXINTR** and **UARTTXINTR** have been separated from the status interrupts. This enables you to use **UARTRXINTR** and **UARTTXINTR** so that data can be read or written in response to the FIFO trigger levels.

The error interrupt, **UARTEINTR**, can be triggered when there is an error in the reception of data. A number of error conditions are possible.

The modem status interrupt, **UARTMSINTR**, is a combined interrupt of all the individual modem status signals.

The status of the individual interrupt sources can be read either from the Raw Interrupt Status Register, UARTRIS or from the Masked Interrupt Status Register, UARMIS.

UARTMSINTR

The modem status interrupt is asserted if any of the modem status signals(**nUARTCTS**, **nUARTDCD**, **nUARTDSR**, and **nUARTRI**) change. It is cleared by writing a 1 to the corresponding bit(s) in the Interrupt Clear Register, UARTICR, depending on the modem status signals that generated the interrupt.

UARTRXINTR

The receive interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the receive FIFO reaches the programmed triggerlevel. When this happens, the receive interrupt is asserted HIGH. The receiveinterrupt is cleared by reading data from the receive FIFO until it becomes lessthan the trigger level, or by clearing the interrupt.
- If the FIFOs are disabled (have a depth of one location) and data is receivedthereby filling the location, the receive interrupt is asserted HIGH. The receiveinterrupt is cleared by performing a single read of the receive FIFO, or by clearingthe interrupt.

UARTTXINTR

The transmit interrupt changes state when one of the following events occurs:

- If the FIFOs are enabled and the transmit FIFO is equal to or lower than theprogrammed trigger level then the transmit interrupt is asserted HIGH. Thetransmit interrupt is cleared by writing data to the transmit FIFO until it becomesgreater than the trigger level, or by clearing the interrupt.
- If the FIFOs are disabled (have a depth of one location) and there is no datapresent in the transmitters single location, the transmit interrupt is asserted HIGH.It is cleared by performing a single write to the transmit FIFO, or by clearing theinterrupt.

To update the transmit FIFO you must:

- Write data to the transmit FIFO, either prior to enabling the UART and theinterrupts, or after enabling the UART and interrupts.

<Note>

The transmit interrupt is based on a transition through a level, rather than on the levelitself. When the interrupt and the UART is enabled before any data is written to thetransmit FIFO the interrupt is not set. The interrupt is only set, after written data leavesthe single location of the transmit FIFO and it becomes empty.

UARTRTINTR

The receive timeout interrupt is asserted when the receive FIFO is not empty, and nomore data is received during a 32-bit period. The receive timeout interrupt is clearedeither when the FIFO becomes empty through reading all the data (or by reading theholding register), or when a 1 is written to the corresponding bit of the Interrupt ClearRegister, UARTICR.

UARTEINTR

The error interrupt is asserted when an error occurs in the reception of data by theUART. The interrupt can be caused by a number of different error conditions:

- framing
- parity
- break
- overrun.

You can determine the cause of the interrupt by reading the Raw Interrupt StatusRegister, UARTRIS or the Masked Interrupt Status Register, UARTMIS. It can be cleared by writing to the relevant bits of the Interrupt ClearRegister, UARTICR (bits 7 to 10 are the error clear bits).

UARTINTR

The interrupts are also combined into a single output, that is an OR function of the individual masked sources. You can connect this output to a system interrupt controller to provide another level of masking on a individual peripheral basis.

The combined UART interrupt is asserted if any of the individual interrupts are asserted and enabled.

24.2.9 ISO-7816

Figure 24-7 represents ISO-7816 interface. UARTTXD and UARTRXD signals are connected to external pads through Smart card Adapter block. Users can select whether to use ISO-7816 through USESMC, SMCTXENB,SMCRXENB signals, which are controllable by register.

In the case of communicating through ISO-7816 interface, SMC pads are controlled by PAD interface of NXOpenDrainAdapter.

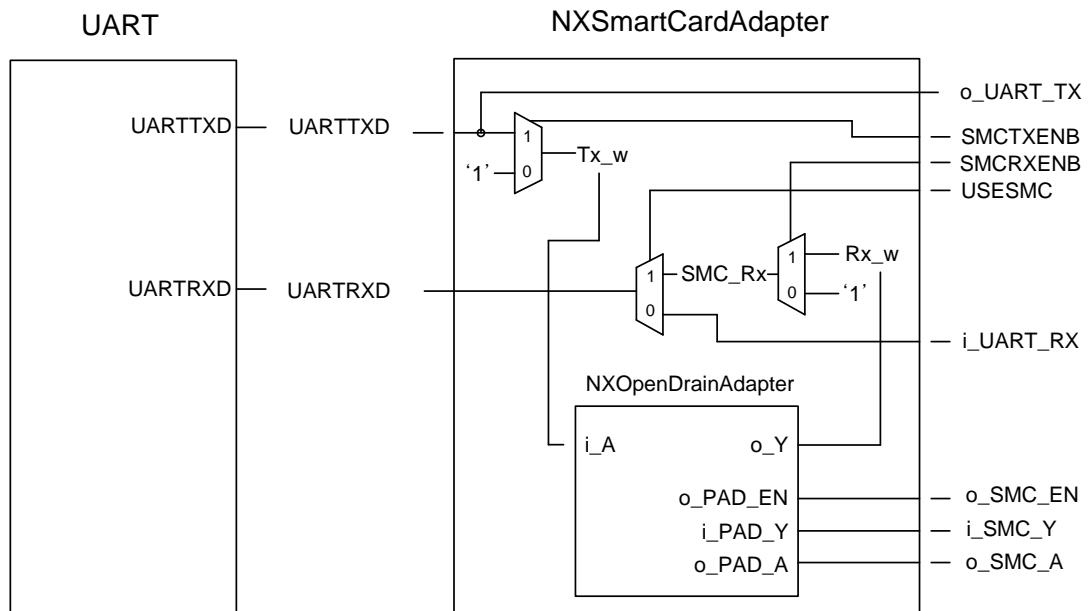


Figure 24-7. Smart Card Adapter Interface

24.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
Data Register (UARTDR)				
The UARTDR Register is the data register. For words to be transmitted: if the FIFOs are enabled, data written to this location is pushed onto the transmitFIFO if the FIFOs are not enabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). The write operation initiates transmission from the UART. The data is prefixed with a start bit, appended with the appropriate parity bit (if parity is enabled), and a stop bit. The resultant word is then transmitted. For received words: if the FIFOs are enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO if the FIFOs are not enabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO).				
Address UART0 : C00A0000 / UART1 : C00A1000 / UART2 : C00A2000 / UART3 : C00A3000 / UART4 : C006D000 / UART5 : C006F000				
[31:12]	-	RESERVED	Reserved	-
[11]	R	OE	Overrun error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0 once there is an empty space in the FIFO and a new character can be written to it.	1'b0
[10]	R	BE	Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state), and the next valid start bit is received.	1'b0
[9]	R	PE	Parity error. When set to 1, it indicates that the parity of the received data character does not match the parity that the EPS and SPS bits in the Line Control Register, UARTLCR_H select.In FIFO mode, this error is associated with the character at the top of the FIFO.	1'b0
[8]	R	FE	Framing error. When set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1). In FIFO mode, this error is associated with the character at the top of the FIFO.	1'b0
[7:0]	R/W	DATA	Receive (read) data character. Transmit (write) data character.	8'b0
Receive Status Register/Error Clear Register (UARTRSR/UARTECR)				
The UARTRSR/UARTECR Register is the receive status register/error clear register. Receive status can also be read from the UARTRSR Register. If the status is read from this register, then the status information for break, framing and parity corresponds to the data character read from the Data Register, UARTDR prior to reading the UARTRSR Register. The status information for overrun is set immediately when an overrun condition occurs. A write to the UARTECR Register clears the framing, parity, break, and overrun errors. All the bits are cleared to 0 on reset. Address UART0:C00A0004 / UART1:C00A1004 / UART2:C00A2004 / UART3:C00A3004 / UART4:C006D004 / UART5:C006F004				
[31:15]	-	RESERVED	Reserved	-
[7:0]	W	-	A write to this register clears the framing, parity, break, and overrun errors. The data value is not important.	-
[31:4]	-	RESERVED	Reserved	-
[3]	R	OE	Overrun error. This bit is set to 1 if data is received and the FIFO is already full. This bit is cleared to 0 by a write to UARTECR. The FIFO contents remain valid because no more data is written when the FIFO is full, only the	1'b0

Bit	R/W	Symbol	Description	Reset Value
			contents of the shift register are overwritten. The CPU must now read the data, to empty the FIFO.	
[2]	R	BE	Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity, and stop bits). This bit is cleared to 0 after a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received.	1b0
[1]	R	PE	Parity error. When set to 1, it indicates that the parity of the received data character does not match the parity that the EPS and SPS bits in the Line Control Register, UARTLCR_H select. This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO.	1b0
[0]	R	FE	Framing error. When set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1). This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO.	1b0
Reserved				
Address UART0 : C00A0008~C00A0014 / UART1 : C00A1008~C00A1014 / UART2 : C00A2008~C00A2014 / UART3 : C00A3008~C00A3014 / UART4 : C006D008~C006D014 / UART5 : C006F008~C006F014				
[31:15]	-	RESERVED	Reserved	-
Flag Register (UARTFR)				
The UARTFR Register is the flag register. After reset TXFF, RXFF, and BUSY are 0, and TXFE and RXFE are 1.				
Address UART0: C00A0018 / UART1: C00A1018 / UART2: C00A2018 / UART3: C00A3018 / UART4: C006D018 / UART5: C006F018				
[31:9]	-	RESERVED	Reserved	-
[8]	R	RI	Ring indicator. This bit is the complement of the UART ring indicator, nUARTRI, modem status input. That is, the bit is 1 when nUARTRI is LOW.	-
[7]	R	TXFF	Transmit FIFO empty. The meaning of this bit depends on the state of the FEN bit in the Line Control Register, UARTLCR_H. If the FIFO is disabled, this bit is set when the transmit holding register is empty. If the FIFO is enabled, the TXFE bit is set when the transmit FIFO is empty. This bit does not indicate if there is data in the transmit shift register.	-
[6]	R	RXFF	Receive FIFO full. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H Register. If the FIFO is disabled, this bit is set when the receive holding register is full. If the FIFO is enabled, the RXFF bit is set when the receive FIFO is full.	-
[5]	R	TXFE	Transmit FIFO full. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H Register. If the FIFO is disabled, this bit is set when the transmit holding register is full. If the FIFO is enabled, the TXFF bit is set when the transmit FIFO is full.	-
[4]	R	RXFE	Receive FIFO empty. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H Register. If the FIFO is disabled, this bit is set when the receive holding register is empty. If the FIFO is enabled, the RXFE bit is set when the receive FIFO is empty.	-
[3]	R	BUSY	UART busy. If this bit is set to 1, the UART is busy transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register.	-

Bit	R/W	Symbol	Description	Reset Value
			This bit is set as soon as the transmit FIFO becomes non-empty, regardless of whether the UART is enabled or not.	
[2]	R	DSD	Data carrier detect. This bit is the complement of the UART data carrier detect, nUARTDCD, modemstatus input. That is, the bit is 1 when nUARTDCD is LOW.	-
[1]	R	DSR	Data set ready. This bit is the complement of the UART data set ready, nUARTDSR, modem statusinput. That is, the bit is 1 when nUARTDSR is LOW.	-
[0]	R	CTS	Clear to send. This bit is the complement of the UART clear to send, nUARTCTS, modem status input. That is, the bit is 1 when nUARTCTS is LOW.	-
Reserved				
Address				
UART0: C00A001C / UART1: C00A101C / UART2: C00A201C / UART3: C00A301C / UART4: C006D01C / UART5: C006F01C				
[31:15]	-	RESERVED	Reserved	-
IrDA Low-Power Counter Register (UARTILPR)				
The UARTILPR Register is the IrDA low-power counter register. This is an 8-bitread/write register that stores the low-power counter divisor value used to generate theirLPBaud16 signal by dividing down of UARTCLK.				
The IrLPBaud16 signal is generated by dividing down the UARTCLK signal according to the low-power divisor value written to the UARTILPR Register.				
The low-power divisor value is calculated as follows:				
low-power divisor (ILPDVSR) = ($F_{UARTCLK} / F_{LPBaud16}$)				
where $F_{LPBaud16}$ is nominally 1.8432MHz.				
You must select the divisor so that $1.42MHz < F_{LPBaud16} < 2.12MHz$, results in a low-power pulse duration of $1.41\text{-}2.11\mu s$ (three times the period of IrLPBaud16).				
Address				
UART0: C00A0020 / UART1: C00A1020 / UART2: C00A2020 / UART3: C00A3020 / UART4: C006D020 / UART5: C006F020				
[31:8]	-	RESERVED	Reserved	-
[7:0]	R/W	ILPDVSR	8-bit low-power divisor value. These bits are cleared to 0 at reset.	8'b0
Integer Baud Rate Register (UARTIBRD)				
The UARTIBRD Register is the integer part of the baud rate divisor value.				
Address				
UART0: C00A0024 / UART1: C00A1024 / UART2: C00A2024 / UART3: C00A3024 / UART4: C006D024 / UART5: C006F024				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	BAUD DIVINT	The integer baud rate divisor. These bits are cleared to 0 on reset.	16'b0
Fractional Baud Rate Register (UARTFBRD)				
The UARTFBRD Register is the fractional part of the baud rate divisor value.				
The baud rate divisor is calculated as follows:				
Baud rate divisor BAUDDIV = ($F_{UARTCLK} / (16 \times \text{Baud rate})$)				
where $F_{UARTCLK}$ is the UART reference clock frequency.				
The BAUDDIV is comprised of the integer value (BAUD DIVINT) and the fractional value (BAUD DIVFRAC).				
Address				
UART0: C00A0028 / UART1: C00A1028 / UART2: C00A2028 / UART3: C00A3028 / UART4: C006D028 / UART5: C006F028				
[31:6]	-	RESERVED	Reserved	-
[5:0]	R/W	BAUD DIVFRAC	The integer baud rate divisor. These bits are cleared to 0 on reset.	6'b0
Line Control Register (UARTLCR_H)				

Bit	R/W	Symbol	Description	Reset Value
The UARTLCR_H Register is the line control register. This register accesses bits 29 to 22 of the UART Line Control Register, UARTLCR.				
All the bits are cleared to 0 when reset.				
The UARTLCR_H, UARTIBRD, and UARTRFBRD registers form the single 30-bitwide UARTLCR Register that is updated on a single write strobe generated by aUARTLCR_H write. So, to internally update the contents of UARTIBRD or UARTRFBRD, aUARTLCR_H write must always be performed at the end.				
To update the three registers there are two possible sequences:				
UARTIBRD write, UARTRFBRD write, and UARTLCR_H write				
UARTRFBRD write, UARTIBRD write, and UARTLCR_H write.				
To update UARTIBRD or UARTRFBRD only:				
UARTIBRD write, or UARTRFBRD write, and UARTLCR_H write.				
Address				
UART0: C00A002C / UART1: C00A102C / UART2: C00A202C / UART3: C00A302C / UART4: C006D02C / UART5: C006F02C				
[31:8]	-	RESERVED	Reserved	-
[7]	R/W	SPS	<p>Stick parity select. 0 = stick parity is disabled 1 = either:</p> <ul style="list-style-type: none"> • if the EPS bit is 0 then the parity bit is transmitted and checked as a 1 • if the EPS bit is 1 then the parity bit is transmitted and checked as a 0. <p>This bit has no effect when the PEN bit disables parity checking and generation.</p>	1b0
[6:5]	R/W	WLEN	<p>Word length. These bits indicate the number of data bits transmitted or received in a frame as follows: b11 = 8 bits b10 = 7 bits b01 = 6 bits b00 = 5 bits.</p>	2b0
[4]	R/W	FEN	<p>Enable FIFOs: 0 = FIFOs are disabled (character mode) that is, the FIFOs become 1-byte-deep holding registers 1 = transmit and receive FIFO buffers are enabled (FIFO mode).</p>	1b0
[3]	R/W	STP2	<p>Two stop bits select. If this bit is set to 1, two stop bits are transmitted at the end of the frame. The receive logic does not check for two stop bits being received.</p>	1b0
[2]	R/W	EPS	<p>Even parity select. Controls the type of parity the UART uses during transmission and reception: 0 = odd parity. The UART generates or checks for an odd number of 1s in the data and parity bits. 1 = even parity. The UART generates or checks for an even number of 1s in the data and parity bits. This bit has no effect when the PEN bit disables parity checking and generation.</p>	1b0
[1]	R/W	PEN	<p>Parity enable: 0 = parity is disabled and no parity bit added to the data frame 1 = parity checking and generation is enabled.</p>	1b0
[0]	R/W	BRK	<p>Send break. If this bit is set to 1, a low-level is continually output on the UARTTXD output, after completing transmission of the current character. For the proper execution of the break command, the software must set this bit for at least two complete frames. For normal use, this bit must be cleared to 0.</p>	1b0
Control Register (UARTCR)				
The UARTCR Register is the control register. All the bits are cleared to 0 on reset except for bits 9 and 8 that are set to 1.				
Address				
UART0: C00A0030 / UART1: C00A1030 / UART2: C00A2030 / UART3: C00A3030 / UART4: C006D030 / UART5: C006F030				
[31:16]	-	RESERVED	Reserved	-

Bit	R/W	Symbol	Description	Reset Value
[15]	R/W	CTSEN	CTS hardware flow control enable. If this bit is set to 1, CTS hardware flow control is enabled. Data is only transmitted when the nUARTCTS signal is asserted.	1'b0
[14]	R/W	RTSEN	RTS hardware flow control enable. If this bit is set to 1, RTS hardware flow control is enabled. Data is only requested when there is space in the receive FIFO for it to be received.	1'b0
[13]	R/W	OUT2	This bit is the complement of the UART Out2 (nUARTOut2) modem status output. That is, when the bit is programmed to a 1, the output is 0. For DTE this can be used as Ring Indicator (RI).	1'b0
[12]	R/W	OUT1	This bit is the complement of the UART Out1 (nUARTOut1) modem status output. That is, when the bit is programmed to a 1 the output is 0. For DTE this can be used as Data Carrier Detect (DCD).	1'b0
[11]	R/W	RTS	Request to send. This bit is the complement of the UART request to send, nUARTRTS, modem status output. That is, when the bit is programmed to a 1 then nUARTRTS is LOW.	1'b0
[10]	R/W	DTR	Data transmit ready. This bit is the complement of the UART data transmit ready, nUARTDTR, modem status output. That is, when the bit is programmed to a 1 then nUARTDTR is LOW.	1'b0
[9]	R/W	RXE	Receive enable. If this bit is set to 1, the receive section of the UART is enabled. Data reception occurs for either UART signals or SIR signals depending on the setting of the SIREN bit. When the UART is disabled in the middle of reception, it completes the current character before stopping.	1'b0
[8]	R/W	TXE	Transmit enable. If this bit is set to 1, the transmit section of the UART is enabled. Data transmission occurs for either UART signals, or SIR signals depending on the setting of the SIREN bit. When the UART is disabled in the middle of transmission, it completes the current character before stopping.	1'b1
[7]	R/W	LBE	<p>Loopback enable. If this bit is set to 1 and the SIREN bit is set to 1 and the SIRTEST bit in the TestControl Register, UARTTCR is set to 1, then the nSIROUT path is inverted, and fed through to the SIRIN path. The SIRTEST bit in the test register must be set to 1 to override the normal half-duplex SIR operation. This must be the requirement for accessing the test registers during normal operation, and SIRTEST must be cleared to 0 when loopback testing is finished. This feature reduces the amount of external coupling required during system test.</p> <p>If this bit is set to 1, and the SIRTEST bit is set to 0, the UARTTXD path is fed through to the UARTRXD path.</p> <p>In either SIR mode or UART mode, when this bit is set, the modem outputs are also fed through to the modem inputs.</p> <p>This bit is cleared to 0 on reset, to disable loopback.</p>	1'b1
[6:3]	-	RESERVED	Reserved	-
[2]	R/W	SIRLP	SIR low-power IrDA mode. This bit selects the IrDA encoding mode. If this bit is cleared to 0, low-level bits are transmitted as an active high pulse with a width of 3/16th of the bit period. If this bit is set to 1, low-level bits are transmitted with a pulse width that is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate. Setting this bit uses less power, but might reduce transmission distances.	1'b0
[1]	R/W	SIREN	<p>SIR enable:</p> <p>0 = IrDA SIR ENDEC is disabled. nSIROUT remains LOW (no light pulse generated), and signal transitions on SIRIN have no effect.</p> <p>1 = IrDA SIR ENDEC is enabled. Data is transmitted and received on nSIROUT and SIRIN. UARTTXD remains HIGH, in the marking state. Signal transitions on UARTRXD or modem status inputs have no effect.</p> <p>This bit has no effect if the UARTEN bit disables the UART.</p>	1'b0
[0]	R/W	UARTEN	<p>UART enable:</p> <p>0 = UART is disabled. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.</p> <p>1 = the UART is enabled. Data transmission and reception occurs for either UART signals or SIR signals depending on the setting of the SIREN bit.</p>	1'b0

Bit	R/W	Symbol	Description	Reset Value
Interrupt FIFO Level Select Register (UARTIFLS)				
The UARTIFLS Register is the interrupt FIFO level select register. You can use this register to define the FIFO level that triggers the assertion of UARTTXINTR and UARTRXINTR.				
The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level.				
The bits are reset so that the trigger level is when the FIFOs are at the half-way mark.				
Address				
UART0: C00A0034 / UART1: C00A1034 / UART2: C00A2034 / UART3: C00A3034 / UART4: C006D034 / UART5: C006F034				
[31:15]	-	RESERVED	Reserved	-
[5:3]	R/W	RXIFLSEL	Receive interrupt FIFO level select. The trigger points for the receive interrupt are as follows: b000 = Receive FIFO becomes \geq 1/8 full b001 = Receive FIFO becomes \geq 1/4 full b010 = Receive FIFO becomes \geq 1/2 full b011 = Receive FIFO becomes \geq 3/4 full b100 = Receive FIFO becomes \geq 7/8 full b101-b111 = reserved.	3'b010
[2:0]	R/W	TXIFLSEL	Transmit interrupt FIFO level select. The trigger points for the transmit interrupt are as follows: b000 = Transmit FIFO becomes \leq 1/8 full b001 = Transmit FIFO becomes \leq 1/4 full b010 = Transmit FIFO becomes \leq 1/2 full b011 = Transmit FIFO becomes \leq 3/4 full b100 = Transmit FIFO becomes \leq 7/8 full b101-b111 = reserved.	3'b010
Interrupt Mask Set/Clear Register (UARTIMSC)				
The UARTIMSC Register is the interrupt mask set/clear register. It is a read/write register.				
On a read this register returns the current value of the mask on the relevant interrupt. On a write of 1 to the particular bit, it sets the corresponding mask of that interrupt. A write of 0 clears the corresponding mask.				
All the bits are cleared to 0 when reset.				
Address				
UART0: C00A0038 / UART1: C00A1038 / UART2: C00A2038 / UART3: C00A3038 / UART4: C006D038 / UART5: C006F038				
[31:11]	-	RESERVED	Reserved	-
[10]	R/W	OEIM	Overrun error interrupt mask. A read returns the current mask for the UARTOEINTR interrupt. On a write of 1, the mask of the UARTOEINTR interrupt is set. A write of 0 clears the mask.	1'b0
[9]	R/W	BEIM	Break error interrupt mask. A read returns the current mask for the UARTBEINTR interrupt. On a write of 1, the mask of the UARTBEINTR interrupt is set. A write of 0 clears the mask.	1'b0
[8]	R/W	PEIM	Parity error interrupt mask. A read returns the current mask for the UARTPEINTR interrupt. On a write of 1, the mask of the UARTPEINTR interrupt is set. A write of 0 clears the mask.	1'b0
[7]	R/W	FEIM	Framing error interrupt mask. A read returns the current mask for the UARTFEINTR interrupt. On a write of 1, the mask of the UARTFEINTR interrupt is set. A write of 0 clears the mask.	1'b0
[6]	R/W	RTIM	Receive timeout interrupt mask. A read returns the current mask for the UARTRTINTR interrupt. On a write of 1, the mask of the UARTRTINTR interrupt is set. A write of 0 clears the mask.	1'b0
[5]	R/W	TXIM	Transmit interrupt mask. A read returns the current mask for the UARTTXINTR interrupt. On a write of 1, the mask of the UARTTXINTR interrupt is set. A write of 0 clears the mask.	1'b0
[4]	R/W	RXIM	Receive interrupt mask. A read returns the current mask for the UARTRXINTR interrupt.	1'b0

Bit	R/W	Symbol	Description	Reset Value
			On a write of 1, the mask of the UARTRXINTR interrupt is set. A write of 0 clears the mask.	
[3]	R/W	DSRMIM	nUARTDSR modem interrupt mask. A read returns the current mask for the UARTDSRINTR interrupt. On a write of 1, the mask of the UARTDSRINTR interrupt is set. A write of 0 clears the mask.	1b0
[2]	R/W	DCDMIM	nUARTDCD modem interrupt mask. A read returns the current mask for the UARTDCDINTR interrupt. On a write of 1, the mask of the UARTDCDINTR interrupt is set. A write of 0 clears the mask.	1b0
[1]	R/W	CTSMIM	nUARTCTS modem interrupt mask. A read returns the current mask for the UARTCTSINTR interrupt. On a write of 1, the mask of the UARTCTSINTR interrupt is set. A write of 0 clears the mask.	1b0
[0]	R/W	RIMIM	nUARTRI modem interrupt mask. A read returns the current mask for the UARTRINTR interrupt. On a write of 1, the mask of the UARTRINTR interrupt is set. A write of 0 clears the mask.	1b0

Interrupt Status Register (UARTRIS)

The UARTRIS Register is the raw interrupt status register. It is a read-only register. This register returns the current raw status value, prior to masking, of the corresponding interrupt. A write has no effect.

All the bits, except for the modem status interrupt bits (bits 3 to 0), are cleared to 0 when reset. The modem status interrupt bits are undefined after reset.

Address

UART0: C00A003C / UART1: C00A103C / UART2: C00A203C / UART3: C00A303C / UART4: C006D03C / UART5: C006F03C

[31:11]	-	RESERVED	Reserved	-
[10]	R	OERIS	Overrun error interrupt status. Returns the raw interrupt state of the UARTOEINTR interrupt.	1b0
[9]	R	BERIS	Break error interrupt status. Returns the raw interrupt state of the UARTBEINTR interrupt.	1b0
[8]	R	PERIS	Parity error interrupt status. Returns the raw interrupt state of the UARTPEINTR interrupt.	1b0
[7]	R	FERIS	Framing error interrupt status. Returns the raw interrupt state of the UARTFEINTR interrupt.	1b0
[6]	R	RTRIS	Receive timeout interrupt status. Returns the raw interrupt state of the UARTRTINTR interrupt. In this case the raw interrupt cannot be set unless the mask is set, this is because the mask acts as an enable for power saving. That is, the same status can be read from UARTRMIS and UARTRIS for the receive timeout interrupt.	1b0
[5]	R	TXRIS	Transmit interrupt status. Returns the raw interrupt state of the UARTRXINTR interrupt.	1b0
[4]	R	RXRIS	Receive interrupt status. Returns the raw interrupt state of the UARTRXINTR interrupt.	1b0
[3]	R	DSRRMIS	nUARTDSR modem interrupt status. Returns the raw interrupt state of the UARTDSRINTR interrupt.	1b0
[2]	R	DCDRMIS	nUARTDCD modem interrupt status. Returns the raw interrupt state of the UARTDCDINTR interrupt.	1b0
[1]	R	CTSRMIS	nUARTCTS modem interrupt status. Returns the raw interrupt state of the UARTCTSINTR interrupt.	1b0
[0]	R	RIRMIS	nUARTRI modem interrupt status. Returns the raw interrupt state of the UARTRINTR interrupt.	1b0

Interrupt Status Register (UARTRMIS)

The UARTRMIS Register is the masked interrupt status register. It is a read-only register. This register returns the current masked status value of the corresponding interrupt. A write has no effect.

All the bits, except for the modem status interrupt bits (bits 3 to 0), are cleared to 0 when reset. The modem status interrupt bits are undefined after reset.

Address

UART0: C00A0040 / UART1: C00A1040 / UART2: C00A2040 / UART3: C00A3040 / UART4: C006D040 / UART5: C006F040

[31:11]	-	RESERVED	Reserved	-
[10]	R	OEMIS	Overrun error masked interrupt status. Returns the masked interrupt state of the UARTOEINTR interrupt.	1b0

Bit	R/W	Symbol	Description	Reset Value
			UARTOEINTR interrupt.	
[9]	R	BEMIS	Break error masked interrupt status. Returns the masked interrupt state of the UARTBEINTR interrupt.	1'b0
[8]	R	PEMIS	Parity error masked interrupt status. Returns the masked interrupt state of the UARTPEINTR interrupt.	1'b0
[7]	R	FEMIS	Framing error masked interrupt status. Returns the masked interrupt state of the UARTFEINTR interrupt.	1'b0
[6]	R	RTMIS	Framing error masked interrupt status. Returns the masked interrupt state of the UARTFEINTR interrupt.	1'b0
[5]	R	TXMIS	Transmit masked interrupt status. Returns the masked interrupt state of the UARTRXINTR interrupt.	1'b0
[4]	R	RXMIS	Receive masked interrupt status. Returns the masked interrupt state of the UARTRXINTR interrupt.	1'b0
[3]	R	DSRMMI	nUARTDSR modem masked interrupt status. Returns the masked interrupt state of the UARTDSRINTR interrupt.	1'b0
[2]	R	DCDMM	nUARTDCD modem masked interrupt status. Returns the masked interrupt state of the UARTDCDINTR interrupt.	1'b0
[1]	R	CTSMMIS	nUARTCTS modem masked interrupt status. Returns the masked interrupt state of the UARTCTSINTR interrupt.	1'b0
[0]	R	RIMMIS	nUARTRI modem masked interrupt status. Returns the masked interrupt state of the UARTRIINTR interrupt.	1'b0

Interrupt Clear Register (UARTICR)

The UARTICR Register is the interrupt clear register and is write-only. On a write of 1, the corresponding interrupt is cleared. A write of 0 has no effect.

Address

UART0: C00A0044 / UART1: C00A1044 / UART2: C00A2044 / UART3: C00A3044 / UART4: C006D044 / UART5: C006F044

[31:15]	-	RESERVED	Reserved	-
[10]	W	OEIC	Overrun error interrupt clear. Clears the UARTOEINTR interrupt.	-
[9]	W	BEIC	Break error interrupt clear. Clears the UARTBEINTR interrupt.	-
[8]	W	PEIC	Parity error interrupt clear. Clears the UARTPEINTR interrupt.	-
[7]	W	FEIC	Framing error interrupt clear. Clears the UARTFEINTR interrupt.	-
[6]	W	RTIC	Receive timeout interrupt clear. Clears the UARTRTINTR interrupt.	-
[5]	W	TXIC	Transmit interrupt clear. Clears the UARTRXINTR interrupt.	-
[4]	W	RXIC	Receive interrupt clear. Clears the UARTRXINTR interrupt.	-
[3]	W	DSRMIC	nUARTDSR modem interrupt clear. Clears the UARTDSRINTR interrupt.	-
[2]	W	DCDMIC	nUARTDCD modem interrupt clear. Clears the UARTDCDINTR interrupt.	-
[1]	W	CTSMIC	nUARTCTS modem interrupt clear. Clears the UARTCTSINTR interrupt.	-
[0]	W	RIMIC	nUARTRI modem interrupt clear. Clears the UARTRIINTR interrupt.	-

DMA Control Register (UARTDMACR)

The UARTDMACR Register is the DMA control register. It is a read/write register. All the bits are cleared to 0 on reset.

Address

UART0: C00A0048 / UART1: C00A1048 / UART2: C00A2048 / UART3: C00A3048 / UART4: C006D048 / UART5: C006F048

[31:15]	-	RESERVED	Reserved	-
---------	---	-----------------	----------	---

Bit	R/W	Symbol	Description	Reset Value
[2]	R/W	DMAONERR	DMA on error. If this bit is set to 1, the DMA receive request outputs, UARTRXDMASREQ orUARTRXDMABREQ, are disabled when the UART error interrupt is asserted.	1b0
[1]	R/W	TXDMAE	Transmit DMA enable. If this bit is set to 1, DMA for the transmit FIFO is enabled.	1b0
[0]	R/W	RXDMAE	Receive DMA enable. If this bit is set to 1, DMA for the receive FIFO is enabled.	1b0

Section 25. **USB2.0 OTG**

25.1 Overview

USB On-The-Go (OTG) is a Dual-Role Device (DRD) controller, which supports both device and host functions. It is fully compliant with the On-The-Go Supplement to the USB 2.0 Specification, Revision 1.0a. It supports high-speed (HS, 480-Mbps), full-speed (FS, 12-Mbps), and low-speed (LS, 1.5-Mbps, Host only) transfers. HS OTG can be configured as a Host-only or Device-only controller.

25.1.1 Features

The USB2.0 HS OTG features include the following:

- Complies with the On-The-Go Supplement to the USB 2.0 Specification (Revision 1.3a)
- Operates in High-Speed (480 Mbps), Full-Speed (12 Mbps) and Low-Speed (1.5 Mbps, Host only) modes
- Supports UTMI+ Level 3 interface (Revision 1.0)
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports only 32-bit data on the AHB
- 1 Control Endpoint 0 for control transfer
- 16 Device Mode Endpoints including Control Endpoint 0
- Programmable endpoint type: Bulk, Isochronous, or Interrupt
- Programmable IN/ OUT direction
- Supports 16 Host channels

25.1.2 Block Diagram

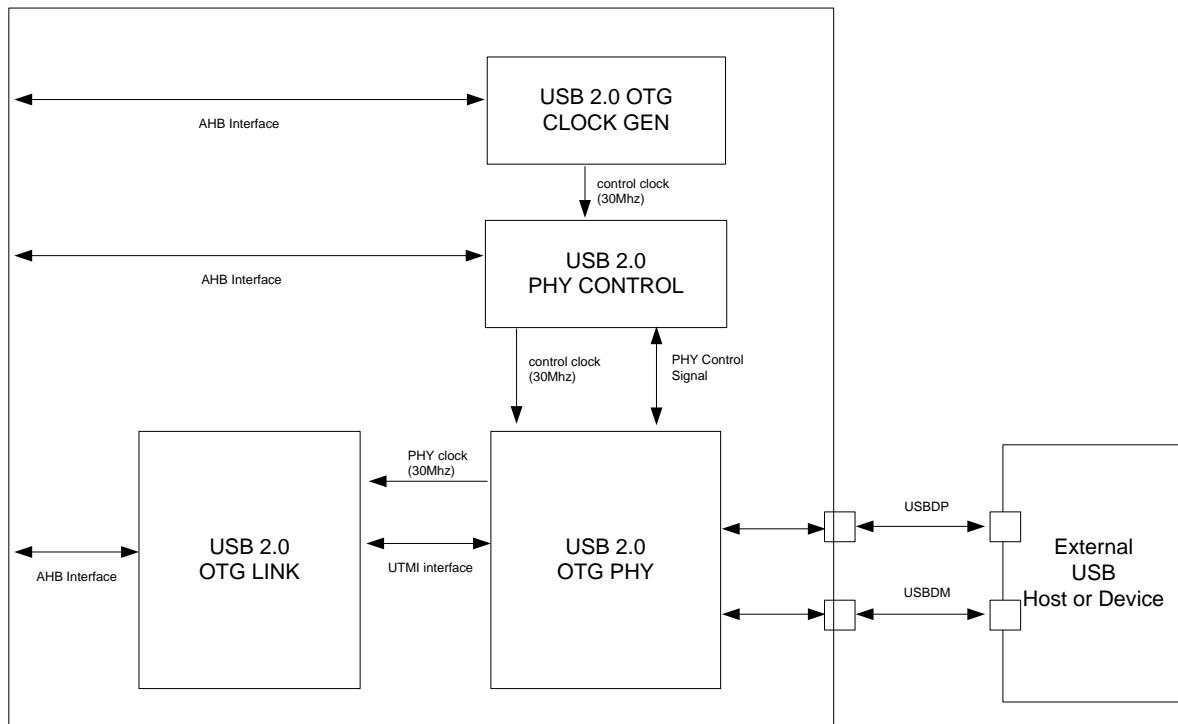


Figure 25-1. USB 2.0 OTG Block Diagram

The blocks in the USB 2.0 OTG controller are comprising as follows:

- USB 2.0 CLOCK GEN
- USB 2.0 PHY Control
- USB 2.0 PHY Link

Each has an AHB Slave, which provides CPU with read and write access to Control and Status Registers. The OTG Link has an AHB Master to transfer data on the AHB.

The USB system shown in Figure 25-1 supports the port:

- USB 2.0 OTG 1 Port

25.1.3 I/O Pin Description

Pin Name	GPIO No.	GPIO Function	Type	Description
USB 2.0 OTG Controller				
USBDP	-	-	I/O	Data Plus Signal from the USB Cable
USBDM	-	-	I/O	Data Minus Signal from the USB Cable
USBREXT	-	-	I	Connection to the external 44.2-ohm(+/- 1%) resistor. The signal must not go through a series resistance in the pad (an ESD resistance is included in the macro). The pad should have ESD, PMOS and NMOS clamp devices. The 44.2-ohm(+/- 1%) resistor must be referenced to the VSSA33C ground supply and placed as close as possible to the chip. Total capacitance should be less than 8pF, including board traces.
USBVBUSIO	-	-	I	USB 5V Power detection This VBUS indicator signal indicates that the VBUS signal on the USB cable is active. For the serial interface, this signal controls the Pull-Up resistance on the D+ line in Device mode only. 1 : Pull-Up resistance on the D+ line is enabled based on the speed of operation 0 : Pull-Up resistance on the D+ line is disable.
USBVBUS	-	-	I	Logic Level VBUS Detect. When VBUSVLDEXTSEL of USB phy register is set to 1, USBVBUS is used instead of USBVBUSIO. This pin is legacy mode.
USBID			IO	USB Mini-Receptacle Identifier. This signal differentiates a mini-A from a mini-B plug. The ID line is sampled only when the PULLUP signal is high. After sampling the ID line, It indicates whether a mini-A or mini-B cable is connected. -Low : Mini-A connected -High: Mini-B connected If this signal is not used, internal resistance pulls the signal's voltage level up to 2.5V.
USBDRVVBUS	GPIOC[11]	Alt3	O	This controller signal enables or disables external charge pump in host mode.

Table 25-1. USB OTG I/O Pin Description

25.2 Functional Description

25.2.1 End Point Packet Size

USB OTG of this chip support 8 end-points. In device mode, maximum packet size per each EP(Endpoint) is as follows:

EP number	Max Packet Size (bytes)
EP[0]	64
EP[1]	512
EP[2]	512
EP[3]	1024
EP[4]	1024
EP[5]	512
EP[6]	512
EP[7]	512
EP[8]	512
EP[9]	512
EP[10]	512
EP[11]	64
EP[12]	64
EP[13]	64
EP[14]	64
EP[15]	64

Table 25-2. maximum packet size per endpoint.

25.2.2 USB 5V Power Detection

You can select one of two VBUS by VBUSVLDEXTSEL of USB phy register.

VBUS pin	Description
USBVBUSIO	USB2.0 OTG module can operate in device mode when the voltage on USBVBUSIO is valid. To be valid in device mode, the voltage on USBVBUSIO is required to be between 4.75V and 5.25V.
USBVBUS	When VBUSVLDEXTSEL of USB phy register is set to 1, USBVBUS is used instead of USBVBUSIO. The USBVBUS pad is a logic-level input. When the USB2 PHY is operating as a device, The USBVBUS acts as a gating signal for many of the internal USB2 PHY modules. Therefore, it is important that transitions on the USBVBUS are clean and well-defined.

25.2.3 Force Power Down

To reduce power consumption, all analog circuits of the USB 2.0 OTG block can be power downed.

See the following table for setting registers to force power down and power up.

operation	Register Setting	Description
Power Up	PHYCTRL.PHYPOR = 0x0237; PHYCTRL.TESTPARM4 = 0x0000;	Power up USB PHY.
Power Down	PHYCTRL.PHYPOR = 0x023C; PHYCTRL.TESTPARM4 = 0x0030;	Power Down USB PHY including : -VBUS valid comparator(Bias and Bandgap circuits) -PLL -XO In power down mode, VBUS signal is gated, and USB PHY does not communicate with external Host/Device regardless of VBUS.

Table 25-3 Force Power down configurations

25.2.4 External Charge Pump

The USB 2.0 PHY requires an off-chip charge pump to provide power to the USB 2.0 OTG PHY VBUS pin.

Figure 25-2 shows the charge pump connection to the USB 2.0 PHY.

The charge pump's output is connected directly to VBUS on the device board. The USB 2.0 PHY's VBUS pin VBUS pin is also connected to VBUS. The charge pump's DRVVBUS input is an output of the OTG HNP finitestate machine and an input to the USB 2.0 PHY.

The VBUS presents a worst-case core-side load of 500 fF. This worst-case load is a small addition to the overall capacitance budget that includes the external capacitive load due to routing, pads, package, board traces, and receivers

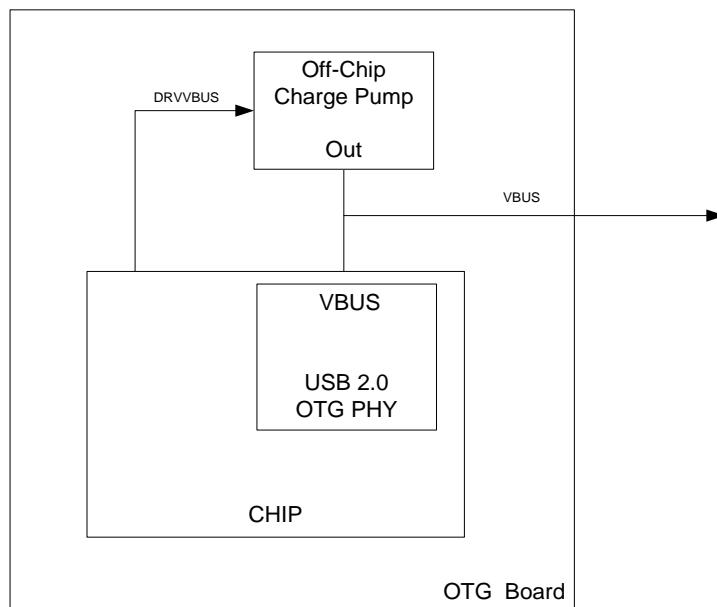


Figure 25-2 Charge Pump Connection

25.2.5 Modes of Operation

The application operates the Link either in DMA mode or in Slave mode. The application cannot operate the core using DMA and Slave modes simultaneously.

- DMA mode

USB OTG host uses the AHB Master interface to transmit packet data fetch (AHB to USB) and receive data update (USB to AHB). The AHB master uses the programmed DMA address (HCDMAN register in Host mode and DIEPDMA_n/ DOEPDMA_n register in Device mode) to access the data buffers.

■ Slave Mode

USB OTG can operate either in transaction-level operation or in pipelined transaction-level operation. The application handles one data packet at a time per channel / endpoint in transaction-level operations. In pipelined transaction-level operation, the application programs the OTG to perform multiple transactions. The advantage of pipelined operation is that the application is not interrupted on packet basis

25.3 Programming User Config of PHY and OTG LINK

- 1) Release otg common reset
 - Program RSTCON1[25](addr: 0xC0012004) to 1'b1
- 2) Program scale mode to real mode
 - Program TIEOFFREG12[1:0](addr: 0xC0011030) to 2'b00
- 3) Select word interface and enable word interface selection
 - 8bit word interface : program TIEOFFREG14[9:8](addr: 0xC0011038) to 2'b01
 - 16bit word interface : program TIEOFFREG14[9:8](addr: 0xC0011038) to 2'b11
- 4) Select VBUS
 - Analog 5V USBVBUSIO : program TIEOFFREG13[25:24](addr: 0xC0011034) to 2'b00
 - Digital USBVBUS : program TIEOFFREG13[25:24](addr: 0xC0011034) to 2'b11
- 5) POR(Power On Reset) of PHY
 - Program TIEOFFREG13[8:7](addr: 0xC0011034) to 2'b01
- 6) Wait clock of PHY – about 40 micro seconds
- 7) Release utmi reset
 - Program TIEOFFREG13[3](addr: 0xC0011034) to 1'b1
- 8) Release ahb reset
 - Program TIEOFFREG13[2](addr: 0xC0011034) to 1'b1

25.4 Register Summary

The OTG Link registers are classified as follows:

- Core Global Registers
- Host Mode Registers
 - Host Global Registers
 - Host Port CSRs
 - Host Channel-Specific Registers
- Device Mode Registers
 - Device Global Registers
 - Device Endpoint-Specific Registers

25.4.1 USB 2.0 OTG Register Map Groups

Symbol	Address	Description
Base address : USB_OTG_BASE = C004_0000H		
USBOTG_GCSR	0xC004_0000	Core Global CSRs (1KB)
USBOTG_HMCSR	0xC004_0400	Host Mode CSRs (1KB)
USBOTG_DMCSR	0xC004_0800	Device Mode CSRs (1.5KB)
USBOTG_PCGCCTL	0xC004_0E00	Power and Clock Gating CSRs (0.5KB)
USBOTG_EP FIFO0	0xC004_1000	Device EP 0 / Host Channel 0 FIFO (4 KB)
USBOTG_EP FIFO1	0xC004_2000	Device EP 1 / Host Channel 1 FIFO (4 KB)
USBOTG_EP FIFO2	0xC004_3000	Device EP 2 / Host Channel 2 FIFO (4 KB)
USBOTG_EP FIFO3	0xC004_4000	Device EP 3 / Host Channel 3 FIFO (4 KB)
USBOTG_EP FIFO4	0xC004_5000	Device EP 4 / Host Channel 4 FIFO (4 KB)
USBOTG_EP FIFO5	0xC004_6000	Device EP 5 / Host Channel 5 FIFO (4 KB)
USBOTG_EP FIFO6	0xC004_7000	Device EP 6 / Host Channel 6 FIFO (4 KB)
USBOTG_EP FIFO7	0xC004_8000	Device EP 7 / Host Channel 7 FIFO (4 KB)
USBOTG_EP FIFO8	0xC004_9000	Device EP 8 / Host Channel 8 FIFO (4 KB)
USBOTG_EP FIFO9	0xC004_A000	Device EP 9 / Host Channel 9 FIFO (4 KB)
USBOTG_EP FIFO10	0xC004_B000	Device EP 10 / Host Channel 10 FIFO (4 KB)
USBOTG_EP FIFO11	0xC004_C000	Device EP 11 / Host Channel 11 FIFO (4 KB)
USBOTG_EP FIFO12	0xC004_D000	Device EP 12 / Host Channel 12 FIFO (4 KB)
USBOTG_EP FIFO13	0xC004_E000	Device EP 13 / Host Channel 13 FIFO (4 KB)
USBOTG_EP FIFO14	0xC004_F000	Device EP 14 / Host Channel 14 FIFO (4 KB)
USBOTG_EP FIFO15	0xC005_0000	Device EP 15 / Host Channel 15 FIFO (4 KB)
reserved	0xC005_1000	
DFIFO_RAM	0xC006_0000	Direct Access to Data FIFO RAM for Debugging (128 KB)

25.4.2 USB 2.0 OTG Register Map Summary

Register	Address	R/W	Description	Reset Value
OTG LINK Core Global Registers <i>BaseAddress : 0xC004_0000</i>				
GOTGCTL	0xC004_0000	R/W	OTG Control and Status Register	0x0001_0000
GOTGINT	0xC004_0004	R/W	OTG Interrupt Register	0x0000_0000
GAHBCFG	0xC004_0008	R/W	Core AHB Configuration Register	0x0000_0000
GUSBCFG	0xC004_000C	R/W	Core USB Configuration Register	0x0000_1400
GRSTCTL	0xC004_0010	R/W	Core Reset Register	0x8000_0000
GINTSTS	0xC004_0014	R/W	Core Interrupt Register	0x0400_1020
GINTMSK	0xC004_0018	R/W	Core Interrupt Mask Register	0x0000_0000
GRXSTSR	0xC004_001C	R	Receive Status Debug Read Register	-
GRXSTSP	0xC004_0020	R	Receive Status Read/Pop Register	-
GRXFISIZ	0xC004_0024	R/W	Receive FIFO Size Register	0x0000_1800
GNPTXFISIZ	0xC004_0028	R/W	Non-Periodic Transmit FIFO Size Register	0x1800_1800
GNPTXSTS	0xC004_002C	R	Non-Periodic Transmit FIFO/Queue Status Register	0x0008_1800
HPTXFISIZ	0xC004_0100	R/W	Host Periodic Transmit FIFO Size Register	0x0300_5A00
DPTXFISIZ1	0xC004_0104	R/W	Device Periodic Transmit FIFO-1 Size Register	0x0300_1000
DPTXFISIZ2	0xC004_0108	R/W	Device Periodic Transmit FIFO-2 Size Register	0x0300_3300
DPTXFISIZ3	0xC004_010C	R/W	Device Periodic Transmit FIFO-3 Size Register	0x0300_3600
DPTXFISIZ4	0xC004_0110	R/W	Device Periodic Transmit FIFO-4 Size Register	0x0300_3900
DPTXFISIZ5	0xC004_0114	R/W	Device Periodic Transmit FIFO-5 Size Register	0x0300_3C00
DPTXFISIZ6	0xC004_0118	R/W	Device Periodic Transmit FIFO-6 Size Register	0x0300_3F00
DPTXFISIZ7	0xC004_011C	R/W	Device Periodic Transmit FIFO-7 Size Register	0x0300_4200
DPTXFISIZ8	0xC004_0120	R/W	Device Periodic Transmit FIFO-8 Size Register	0x0300_4500
DPTXFISIZ9	0xC004_0124	R/W	Device Periodic Transmit FIFO-9 Size Register	0x0300_4800
DPTXFISIZ10	0xC004_0128	R/W	Device Periodic Transmit FIFO-10 Size Register	0x0300_4B00
DPTXFISIZ11	0xC004_012C	R/W	Device Periodic Transmit FIFO-11 Size Register	0x0300_4E00
DPTXFISIZ12	0xC004_0130	R/W	Device Periodic Transmit FIFO-12 Size Register	0x0300_5100
DPTXFISIZ13	0xC004_0134	R/W	Device Periodic Transmit FIFO-13 Size Register	0x0300_5400
DPTXFISIZ14	0xC004_0138	R/W	Device Periodic Transmit FIFO-14 Size Register	0x0300_5700
DPTXFISIZ15	0xC004_013C	R/W	Device Periodic Transmit FIFO-15 Size Register	0x0300_5A00
Host Global Registers (Host Mode)				
HCFG	0xC004_0400	R/W	Host Configuration Register	0x0020_0000
HFIR	0xC004_0404	R/W	Host Frame Interval Register	0x0000_17D7

Register	Address	R/W	Description	Reset Value
HNUM	0xC004_0408	R	Host Frame Number/Frame Time Remaining Register	0x0000_0000
HPTXSTS	0xC004_0410	R	Host Periodic Transmit FIFO/Queue Status Register	0x0008_0100
HAINT	0xC004_0414	R	Host All Channels Interrupt Register	0x0000_0000
HAINTMSK	0xC004_0418	R/W	Host All Channels Interrupt Mask Register	0x0000_0000
Host Port Control and Status Registers(Host Mode)				
HPRT	0xC004_0440	R/W	Host Port Control and Status Register	0x0000_0000
Host Channel-Specific Registers(Host Mode)				
HCCHAR0	0xC004_0500	R/W	Host Channel 0 Characteristics Register	0x0000_0000
HCSPLT0	0xC004_0504	R/W	Host Channel 0 Split Control Register	0x0000_0000
HCINT0	0xC004_0508	R/W	Host Channel 0 Interrupt Register	0x0000_0000
HCINTMSK0	0xC004_050C	R/W	Host Channel 0 Interrupt Mask Register	0x0000_0000
HCTSIZ0	0xC004_0510	R/W	Host Channel 0 Transfer Size Register	0x0000_0000
HCDMA0	0xC004_0514	R/W	Host Channel 0 DMA Address Register	0x0000_0000
HCCHAR1	0xC004_0520	R/W	Host Channel 1 Characteristics Register	0x0000_0000
HCSPLT1	0xC004_0524	R/W	Host Channel 1 Split Control Register	0x0000_0000
HCINT1	0xC004_0528	R/W	Host Channel 1 Interrupt Register	0x0000_0000
HCINTMSK1	0xC004_052C	R/W	Host Channel 1 Interrupt Mask Register	0x0000_0000
HCTSIZ1	0xC004_0530	R/W	Host Channel 1 Transfer Size Register	0x0000_0000
HCDMA1	0xC004_0534	R/W	Host Channel 1 DMA Address Register	0x0000_0000
HCCHAR2	0xC004_0540	R/W	Host Channel 2 Characteristics Register	0x0000_0000
HCSPLT2	0xC004_0544	R/W	Host Channel 2 Split Control Register	0x0000_0000
HCINT2	0xC004_0548	R/W	Host Channel 2 Interrupt Register	0x0000_0000
HCINTMSK2	0xC004_054C	R/W	Host Channel 2 Interrupt Mask Register	0x0000_0000
HCTSIZ2	0xC004_0550	R/W	Host Channel 2 Transfer Size Register	0x0000_0000
HCDMA2	0xC004_0554	R/W	Host Channel 2 DMA Address Register	0x0000_0000
HCCHAR3	0xC004_0560	R/W	Host Channel 3 Characteristics Register	0x0000_0000
HCSPLT3	0xC004_0564	R/W	Host Channel 3 Split Control Register	0x0000_0000
HCINT3	0xC004_0568	R/W	Host Channel 3 Interrupt Register	0x0000_0000
HCINTMSK3	0xC004_056C	R/W	Host Channel 3 Interrupt Mask Register	0x0000_0000
HCTSIZ3	0xC004_0570	R/W	Host Channel 3 Transfer Size Register	0x0000_0000
HCDMA3	0xC004_0574	R/W	Host Channel 3 DMA Address Register	0x0000_0000
HCCHAR4	0xC004_0580	R/W	Host Channel 4 Characteristics Register	0x0000_0000
HCSPLT4	0xC004_0584	R/W	Host Channel 4 Split Control Register	0x0000_0000
HCINT4	0xC004_0588	R/W	Host Channel 4 Interrupt Register	0x0000_0000

Register	Address	R/W	Description	Reset Value
HCINTMSK4	0xC004_058C	R/W	Host Channel 4 Interrupt Mask Register	0x0000_0000
HCTSIZ4	0xC004_0580	R/W	Host Channel 4 Transfer Size Register	0x0000_0000
HCDMA4	0xC004_0584	R/W	Host Channel 4 DMA Address Register	0x0000_0000
HCCHAR5	0xC004_05A0	R/W	Host Channel 5 Characteristics Register	0x0000_0000
HCSPLT5	0xC004_05A4	R/W	Host Channel 5 Split Control Register	0x0000_0000
HCINT5	0xC004_05A8	R/W	Host Channel 5 Interrupt Register	0x0000_0000
HCINTMSK5	0xC004_05AC	R/W	Host Channel 5 Interrupt Mask Register	0x0000_0000
HCTSIZ5	0xC004_05B0	R/W	Host Channel 5 Transfer Size Register	0x0000_0000
HCDMA5	0xC004_05B4	R/W	Host Channel 5 DMA Address Register	0x0000_0000
HCCHAR6	0xC004_05C0	R/W	Host Channel 6 Characteristics Register	0x0000_0000
HCSPLT6	0xC004_05C4	R/W	Host Channel 6 Split Control Register	0x0000_0000
HCINT6	0xC004_05C8	R/W	Host Channel 6 Interrupt Register	0x0000_0000
HCINTMSK6	0xC004_05CC	R/W	Host Channel 6 Interrupt Mask Register	0x0000_0000
HCTSIZ6	0xC004_05D0	R/W	Host Channel 6 Transfer Size Register	0x0000_0000
HCDMA6	0xC004_05D4	R/W	Host Channel 6 DMA Address Register	0x0000_0000
HCCHAR7	0xC004_05E0	R/W	Host Channel 7 Characteristics Register	0x0000_0000
HCSPLT7	0xC004_05E4	R/W	Host Channel 7 Split Control Register	0x0000_0000
HCINT7	0xC004_05E8	R/W	Host Channel 7 Interrupt Register	0x0000_0000
HCINTMSK7	0xC004_05EC	R/W	Host Channel 7 Interrupt Mask Register	0x0000_0000
HCTSIZ7	0xC004_05F0	R/W	Host Channel 7 Transfer Size Register	0x0000_0000
HCDMA7	0xC004_05F4	R/W	Host Channel 7 DMA Address Register	0x0000_0000
HCCHAR8	0xC004_0600	R/W	Host Channel 8 Characteristics Register	0x0000_0000
HCSPLT8	0xC004_0604	R/W	Host Channel 8 Split Control Register	0x0000_0000
HCINT8	0xC004_0608	R/W	Host Channel 8 Interrupt Register	0x0000_0000
HCINTMSK8	0xC004_060C	R/W	Host Channel 8 Interrupt Mask Register	0x0000_0000
HCTSIZ8	0xC004_0610	R/W	Host Channel 8 Transfer Size Register	0x0000_0000
HCDMA8	0xC004_0614	R/W	Host Channel 8 DMA Address Register	0x0000_0000
HCCHAR9	0xC004_0620	R/W	Host Channel 9 Characteristics Register	0x0000_0000
HCSPLT9	0xC004_0624	R/W	Host Channel 9 Split Control Register	0x0000_0000
HCINT9	0xC004_0628	R/W	Host Channel 9 Interrupt Register	0x0000_0000
HCINTMSK9	0xC004_062C	R/W	Host Channel 10 Interrupt Mask Register	0x0000_0000
HCTSIZ9	0xC004_0630	R/W	Host Channel 10 Transfer Size Register	0x0000_0000
HCDMA9	0xC004_0634	R/W	Host Channel 10 DMA Address Register	0x0000_0000
HCCHAR10	0xC004_0640	R/W	Host Channel 10 Characteristics Register	0x0000_0000
HCSPLT10	0xC004_0644	R/W	Host Channel 10 Split Control Register	0x0000_0000

Register	Address	R/W	Description	Reset Value
HCINT10	0xC004_0648	R/W	Host Channel 10Interrupt Register	0x0000_0000
HCINTMSK10	0xC004_064C	R/W	Host Channel 10Interrupt Mask Register	0x0000_0000
HCTSIZ10	0xC004_0650	R/W	Host Channel 10Transfer Size Register	0x0000_0000
HCDMA10	0xC004_0654	R/W	Host Channel 10DMA Address Register	0x0000_0000
HCCHAR11	0xC004_0660	R/W	Host Channel 11Characteristics Register	0x0000_0000
HCSPLT11	0xC004_0664	R/W	Host Channel 11Split Control Register	0x0000_0000
HCINT11	0xC004_0668	R/W	Host Channel 11Interrupt Register	0x0000_0000
HCINTMSK11	0xC004_066C	R/W	Host Channel 11Interrupt Mask Register	0x0000_0000
HCTSIZ11	0xC004_0670	R/W	Host Channel 11Transfer Size Register	0x0000_0000
HCDMA11	0xC004_0674	R/W	Host Channel 11DMA Address Register	0x0000_0000
HCCHAR12	0xC004_0680	R/W	Host Channel 12Characteristics Register	0x0000_0000
HCSPLT12	0xC004_0684	R/W	Host Channel 12Split Control Register	0x0000_0000
HCINT12	0xC004_0688	R/W	Host Channel 12Interrupt Register	0x0000_0000
HCINTMSK12	0xC004_068C	R/W	Host Channel 12Interrupt Mask Register	0x0000_0000
HCTSIZ12	0xC004_0690	R/W	Host Channel 12Transfer Size Register	0x0000_0000
HCDMA12	0xC004_0694	R/W	Host Channel 12DMA Address Register	0x0000_0000
HCCHAR13	0xC004_06A0	R/W	Host Channel 13Characteristics Register	0x0000_0000
HCSPLT13	0xC004_06A4	R/W	Host Channel 13Split Control Register	0x0000_0000
HCINT13	0xC004_06A8	R/W	Host Channel 13Interrupt Register	0x0000_0000
HCINTMSK13	0xC004_06AC	R/W	Host Channel 13Interrupt Mask Register	0x0000_0000
HCTSIZ13	0xC004_06B0	R/W	Host Channel 13Transfer Size Register	0x0000_0000
HCDMA13	0xC004_06B4	R/W	Host Channel 13DMA Address Register	0x0000_0000
HCCHAR14	0xC004_06C0	R/W	Host Channel 14Characteristics Register	0x0000_0000
HCSPLT14	0xC004_06C4	R/W	Host Channel 14Split Control Register	0x0000_0000
HCINT14	0xC004_06C8	R/W	Host Channel 14Interrupt Register	0x0000_0000
HCINTMSK14	0xC004_06CC	R/W	Host Channel 14Interrupt Mask Register	0x0000_0000
HCTSIZ14	0xC004_06D0	R/W	Host Channel 14Transfer Size Register	0x0000_0000
HCDMA14	0xC004_06D4	R/W	Host Channel 14DMA Address Register	0x0000_0000
HCCHAR15	0xC004_06E0	R/W	Host Channel 15Characteristics Register	0x0000_0000
HCSPLT15	0xC004_06E4	R/W	Host Channel 15Split Control Register	0x0000_0000
HCINT15	0xC004_06E8	R/W	Host Channel 15Interrupt Register	0x0000_0000
HCINTMSK15	0xC004_06EC	R/W	Host Channel 15Interrupt Mask Register	0x0000_0000
HCTSIZ15	0xC004_06F0	R/W	Host Channel 15Transfer Size Register	0x0000_0000
HCDMA15	0xC004_06F4	R/W	Host Channel 15DMA Address Register	0x0000_0000
Device Global Registers (Device Mode)				

Register	Address	R/W	Description	Reset Value
DCFG	0xC004_0800	R/W	R/W Device Configuration Register	0x0020_0000
DCTL	0xC004_0804	R/W	Device Control Register	0x0000_0000
DSTS	0xC004_0808	R	Device Status Register	0x0000_0002
DIEPMSK	0xC004_0810	R/W	Device IN Endpoint Common Interrupt Mask Register	0x0000_0000
DOEPMSK	0xC004_0814	R/W	Device OUT Endpoint Common Interrupt Mask Register	0x0000_0000
DAINT	0xC004_0818	R	Device ALL Endpoints Interrupt Register	0x0000_0000
DAINTMSK	0xC004_081C	R/W	Device ALL Endpoints Interrupt Mask Register	0x0000_0000
DTKNQR1	0xC004_0820	R	Device IN Token Sequence Learning Queue Read Register 1	0x0000_0000
DTKNQR2	0xC004_0824	R	Device IN Token Sequence Learning Queue Read Register 2	0x0000_0000
DVBUSDIS	0xC004_0828	R/W	Device VBUS Discharge Time Register	0x0000_17D7
DVBUSPULSE	0xC004_082C	R/W	Device VBUS Pulsing Time Register	0x0000_05B8
DTKNQR3	0xC004_0830	R	Device IN Token Sequence Learning Queue Read Register 3	0x0000_0000
DTKNQR4	0xC004_0834	R	Device IN Token Sequence Learning Queue Read Register 4	0x0000_0000

Device Logical IN Endpoint-Specific Registers (Device Mode)

DIEPCTL0	0xC004_0900	R/W	Device Control IN Endpoint 0 Control Register	0x0000_8000
DIEPINT0	0xC004_0908	R/W	Device IN Endpoint 0 Interrupt Register	0x0000_0000
DIEPTSIZ0	0xC004_0910	R/W	Device IN Endpoint 0 Transfer Size Register	0x0000_0000
DIEPDMA0	0xC004_0914	R/W	Device IN Endpoint 0 DMA Address Register	0x0000_0000
DIEPCTL1	0xC004_0920	R/W	Device Control IN Endpoint 1 Control Register	0x0000_0000
DIEPINT1	0xC004_0928	R/W	Device IN Endpoint 1 Interrupt Register	0x0000_0080
DIEPTSIZ1	0xC004_0930	R/W	Device IN Endpoint 1 Transfer Size Register	0x0000_0000
DIEPDMA1	0xC004_0934	R/W	Device IN Endpoint 1 DMA Address Register	0x0000_0000
DIEPCTL2	0xC004_0940	R/W	Device Control IN Endpoint 2 Control Register	0x0000_0000
DIEPINT2	0xC004_0948	R/W	Device IN Endpoint 2 Interrupt Register	0x0000_0080
DIEPTSIZ2	0xC004_0950	R/W	Device IN Endpoint 2 Transfer Size Register	0x0000_0000
DIEPDMA2	0xC004_0954	R/W	Device IN Endpoint 2 DMA Address Register	0x0000_0000
DIEPCTL3	0xC004_0960	R/W	Device Control IN Endpoint 3 Control Register	0x0000_0000
DIEPINT3	0xC004_0968	R/W	Device IN Endpoint 3 Interrupt Register	0x0000_0080
DIEPTSIZ3	0xC004_0970	R/W	Device IN Endpoint 3 Transfer Size Register	0x0000_0000
DIEPDMA3	0xC004_0974	R/W	Device IN Endpoint 3 DMA Address Register	0x0000_0000
DIEPCTL4	0xC004_0980	R/W	Device Control IN Endpoint 4 Control Register	0x0000_0000
DIEPINT4	0xC004_0988	R/W	Device IN Endpoint 4 Interrupt Register	0x0000_0080
DIEPTSIZ4	0xC004_0990	R/W	Device IN Endpoint 4 Transfer Size Register	0x0000_0000
DIEPDMA4	0xC004_0994	R/W	Device IN Endpoint 4 DMA Address Register	0x0000_0000
DIEPCTL5	0xC004_09A0	R/W	Device Control IN Endpoint 5 Control Register	0x0000_0000

Register	Address	R/W	Description	Reset Value
DIEPINT5	0xC004_09A8	R/W	Device IN Endpoint 5 Interrupt Register	0x0000_0000
DIEPTSIZ5	0xC004_09B0	R/W	Device IN Endpoint 5 Transfer Size Register	0x0000_0000
DIEPDMA5	0xC004_09B4	R/W	Device IN Endpoint 5 DMA Address Register	0x0000_0000
DIEPCTL6	0xC004_09C0	R/W	Device Control IN Endpoint 6 Control Register	0x0000_0000
DIEPINT6	0xC004_09C8	R/W	Device IN Endpoint 6 Interrupt Register	0x0000_0000
DIEPTSIZ6	0xC004_09D0	R/W	Device IN Endpoint 6 Transfer Size Register	0x0000_0000
DIEPDMA6	0xC004_09D4	R/W	Device IN Endpoint 6 DMA Address Register	0x0000_0000
DIEPCTL7	0xC004_09E0	R/W	Device Control IN Endpoint 7 Control Register	0x0000_0000
DIEPINT7	0xC004_09E8	R/W	Device IN Endpoint 7 Interrupt Register	0x0000_0000
DIEPTSIZ7	0xC004_09F0	R/W	Device IN Endpoint 7 Transfer Size Register	0x0000_0000
DIEPDMA7	0xC004_09F4	R/W	Device IN Endpoint 7 DMA Address Register	0x0000_0000

Device Logical OUT Endpoint-Specific Registers (Device Mode)

DOEPCTL0	0xC004_0B00	R/W	Device Control OUT Endpoint 0 Control Register	0x0000_8000
DOEPINT0	0xC004_0B08	R/W	Device OUT Endpoint 0 Interrupt Register	0x0000_0000
DOEPTSIZ0	0xC004_0B10	R/W	Device OUT Endpoint 0 Transfer Size Register	0x0000_0000
DOEPDMA0	0xC004_0B14	R/W	Device OUT Endpoint 0 DMA Address Register	0x0000_0000
DOEPCTL1	0xC004_0B00	R/W	Device Control OUT Endpoint 1 Control Register	0x0000_0000
DOEPINT1	0xC004_0B08	R/W	Device OUT Endpoint 1 Interrupt Register	0x0000_0000
DOEPTSIZ1	0xC004_0B10	R/W	Device OUT Endpoint 1 Transfer Size Register	0x0000_0000
DOEPDMA1	0xC004_0B14	R/W	Device OUT Endpoint 1 DMA Address Register	0x0000_0000
DOEPCTL2	0xC004_0B00	R/W	Device Control OUT Endpoint 2 Control Register	0x0000_0000
DOEPINT2	0xC004_0B08	R/W	Device OUT Endpoint 2 Interrupt Register	0x0000_0000
DOEPTSIZ2	0xC004_0B10	R/W	Device OUT Endpoint 2 Transfer Size Register	0x0000_0000
DOEPDMA2	0xC004_0B14	R/W	Device OUT Endpoint 2 DMA Address Register	0x0000_0000
DOEPCTL3	0xC004_0B00	R/W	Device Control OUT Endpoint 3 Control Register	0x0000_0000
DOEPINT3	0xC004_0B08	R/W	Device OUT Endpoint 3 Interrupt Register	0x0000_0000
DOEPTSIZ3	0xC004_0B10	R/W	Device OUT Endpoint 3 Transfer Size Register	0x0000_0000
DOEPDMA3	0xC004_0B14	R/W	Device OUT Endpoint 3 DMA Address Register	0x0000_0000
DOEPCTL4	0xC004_0B00	R/W	Device Control OUT Endpoint 4 Control Register	0x0000_0000
DOEPINT4	0xC004_0B08	R/W	Device OUT Endpoint 4 Interrupt Register	0x0000_0000
DOEPTSIZ4	0xC004_0B10	R/W	Device OUT Endpoint 4 Transfer Size Register	0x0000_0000
DOEPDMA4	0xC004_0B14	R/W	Device OUT Endpoint 4 DMA Address Register	0x0000_0000
DOEPCTL5	0xC004_0B00	R/W	Device Control OUT Endpoint 5 Control Register	0x0000_0000
DOEPINT5	0xC004_0B08	R/W	Device OUT Endpoint 5 Interrupt Register	0x0000_0000
DOEPTSIZ5	0xC004_0B10	R/W	Device OUT Endpoint 5 Transfer Size Register	0x0000_0000

Register	Address	R/W	Description	Reset Value
DOEPDMA5	0xC004_0B14	R/W	Device OUT Endpoint 5 DMA Address Register	0x0000_0000
DOEPCTL6	0xC004_0B00	R/W	Device Control OUT Endpoint 6 Control Register	0x0000_0000
DOEPINT6	0xC004_0B08	R/W	Device OUT Endpoint 6 Interrupt Register	0x0000_0000
DOEPTSIZ6	0xC004_0B10	R/W	Device OUT Endpoint 6 Transfer Size Register	0x0000_0000
DOEPDMA6	0xC004_0B14	R/W	Device OUT Endpoint 6 DMA Address Register	0x0000_0000
DOEPCTL7	0xC004_0B00	R/W	Device Control OUT Endpoint 7 Control Register	0x0000_0000
DOEPINT7	0xC004_0B08	R/W	Device OUT Endpoint 7 Interrupt Register	0x0000_0000
DOEPTSIZ7	0xC004_0B10	R/W	Device OUT Endpoint 7 Transfer Size Register	0x0000_0000
DOEPDMA7	0xC004_0B14	R/W	Device OUT Endpoint 7 DMA Address Register	0x0000_0000
Power and Clock Gating Register				
PCGCCTL	0xC004_0E00	R/W	Power and Clock Gating Control Register	0x0000_0000

25.4.3 Core Global Registers (USB_OTG_GCSR)

Bit	R/W	Symbol	Description	Reset Value
GOTGCTL Address : 0xC004_0000 Reset Value : 0x00010000				
[31:21]	-	RESERVED	Reserved	-
[20]	R/W	OTGVER	OTG Version Indicates the OTG revision. ■ 1'b0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. ■ 1'b1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.	0x0
[19]	R/W	BSESVLD	B-Session Valid Indicates the Device mode transceiver status. 0 : B-session is not valid 1 : B-session is valid	0x0
[18]	R	ASESVLD	A-Session Valid Indicates the Host mode transceiver status. 0 : A-session is not valid 1 : A-session is valid	0x0
[17]	R	DBNCTIME	Long/ Short Debounce Time Indicates the Debounce time of a detected connection. 0 : Long Debounce time, used for physical connections 1 : Short Debounce time, used for soft connections	0x0
[16]	R	CONIDSTS	Connector ID Status Indicates the connector ID status. 0 : The OTG core is in A-device mode 1 : The OTG core is in B-device mode	0x1
[15:12]	-	RESERVED	Reserved	-

Bit	R/W	Symbol	Description	Reset Value
[11]	R/W	DEVHNPEN	Device HNP Enable The application sets the bit if it successfully receives a SetFeature. 0 : HNP is not enabled in the application 1 : HNP is enabled in the application	0x0
[10]	R/W	HSTSETHNPEN	Host Set HNP Enable The application sets this bit if it has successfully enabled HNP on the connected device. 0 : Host Set HNP is not enabled 1 : Host Set HNP is enabled	0x0
[9]	R/W	HNPREQ	HNP Request The application sets this bit to initiate an HNP request to the connected USB host. The core clears this bit if the HstNegSucStsChng bit is cleared. 0 : No HNP request 1 : HNP request	0x0
[8]	R	HSTNEGSCS	Host Negotiation Success The core sets this bit if host negotiation is successful. The core clears this bit if the HNP Request (HNPReq) bit in this register is set. 0 : Host negotiation failure 1 : Host negotiation success	0x0
[7]	R/W	BVALIDOVAL	B-Peripheral Session Valid OverrideValue This bit is used to set the Override value for Bvalid signal when GOTGCTL.BvalidOvEn is set. <ul style="list-style-type: none">■ 1'b0: Bvalid value is 1'b0 when GOTGCTL.BvalidOvEn = 1.■ 1'b1: Bvalid value is 1'b1 when GOTGCTL.BvalidOvEn = 1.	0x0-
[6]	R/W	BVALIDOVEN	B-Peripheral Session Valid Override Enable This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.BvalidOvVal. <ul style="list-style-type: none">■ 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.BvalidOvVal.■ 1'b0: Override is disabled and Bvalid signal from the respective PHY selected is used internally by the core.	0x0
[5]	R/W	AVALIDOVAL	A-Peripheral Session Valid OverrideValue This bit is used to set the Override value for Avalid signal when GOTGCTL.AvalidOvEn is set. <ul style="list-style-type: none">■ 1'b0: Avalid value is 1'b0 when GOTGCTL.AvalidOvEn = 1.■ 1'b1: Avalid value is 1'b1 when GOTGCTL.AvalidOvEn = 1.	0x0
[4]	R/W	AVALIDOVEN	A-Peripheral Session Valid Override Enable This bit is used to enable/disable the software to override the Avalid signal using the GOTGCTL.AvalidOvVal. <ul style="list-style-type: none">■ 1'b1: Internally Avalid received from the PHY is overridden with GOTGCTL.AvalidOvVal.■ 1'b0: Override is disabled and Avalid signal from the respective PHY is used internally by the core.	0x0
[3]	R/W	VBVALIDOVAL	VBUS Valid OverrideValue This bit is used to set the Override value for vbus valid signal when GOTGCTL.VbusvalidOvEn is set. <ul style="list-style-type: none">■ 1'b0: vbusvalid value is 1'b0 when GOTGCTL.VbvalidOvEn = 1.■ 1'b1: vbusvalid value is 1'b1 when GOTGCTL.VbvalidOvEn = 1.	0x0
[2]	R/W	VBVALIDOVEN	VBUS Valid Override Enable (VbvalidOvEn) This bit is used to enable/disable the software to override the vbus-valid signal using the GOTGCTL.VbvalidOvVal. <ul style="list-style-type: none">■ 1'b1: The vbus-valid signal received from the PHY is overridden with	0x0

Bit	R/W	Symbol	Description	Reset Value
			GOTGCTL.vbvalidOval. ■ 1'b0: Override is disabled and avalid signal from the respective PHY is used internally by the core.	
[1]	R/W	SESREQ	Session Request The application sets this bit to initiate a session request on theUSB. The core clears this bit if the HstNegSucStsChng bit is cleared. 0 : No session request 1 : Session request	0x0
[0]	R-	SESREQSCS	Session Request Success The core sets this bit if a session request initiation is successful. 0 : Session request failure 1 : Session request success	0x0-
GOTGINT Address : 0xC004_0004 Reset Value : 0x00000000				
[31:20]	-	RESERVED	Reserved	-
[19]	R/W	DBNCHEDONE	Debounce Done The core sets this bit if the debounce is complete after thedevice connects. This bit is only valid if the HNP Capable orSRP Capable bit is set in the Core USB Configurationregister.	0x0
[18]	R/W	ADEVTOUTCNG	A-Device Timeout Change The core sets this bit to indicate that the A-device has timedout while waiting for the B-device to connect.	0x0
[17]	R/W	HSTNEGDET	Host Negotiation Detected. The core sets this bit if it detects a host negotiation requeston the USB.	0x0
[16:10]	-	RESERVED	Reserved	-
[9]	R/W	HSTNEGSUCSTSCHNG	Host Negotiation Success Status Change The core sets this bit on the success or failure of aUSB host negotiation request.	0x0
[8]	R/W	SESREQSUCSTSCHNG	Session Request Success Status Change The core sets this bit on the success or failure of a sessionrequest.	0x0
[7:3]	-	RESERVED	Reserved	-
[2]	R/W	SESENDDET	Session End Detected The core sets this bit if the b_valid signal is deasserted.	0x0
[1:0]	-	RESERVED	Reserved	-
GAHBCFG Address : 0xC004_0008 Reset Value : 0x00000000				
[31:9]	-	RESERVED	Reserved	-
[8]	R/W	PTXFEMPLVL	Periodic TxFIFO Empty Level Indicates if the Periodic TxFIFO Empty Interrupt bit in the CoreInterrupt registers (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. 0: GINTSTS.PTxFEmp interrupt indicates that the PeriodicTxFIFO is half empty 1: GINTSTS.PTxFEmp interrupt indicates that the PeriodicTxFIFO is completely empty	0x0
[7]	R/W	NPTXFEMPLVL	Non-Periodic TxFIFO Empty Level Indicates if the Non-Periodic TxFIFO Empty Interrupt bits in theCore Interrupt register	0x0

Bit	R/W	Symbol	Description	Reset Value
			(GINSTS.NPTxFEmp) is triggered. This bit is used only in Slave mode. 0 : GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty 1 : GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is completely empty	
[5]	R/W	DMAEN	DMA Enable 0 : Core operates in Slave mode 1 : Core operates in a DMA mode	0x0
[4:1]	R/W	HBSTLEN	Burst Length/Type Internal DMA Mode - AHB Master burst type: 0 : Single 1 : INCR 3 : INCR4 5 : INCR8 7 : INCR16 Others : Reserved	0x0
[0]	R/W	GLBLINTRMSK	Global Interrupt Mask The application uses this bit to mask or unmask the interrupt line assertion to itself. 0 : Mask the interrupt assertion to the application 1 : Unmask the interrupt assertion to the application	0x0
GUSBCFG Address : 0xC004_000C Reset Value : 0x00001400				
[316]	-	RESERVED	Reserved	-
[30]	R/W	FORCEDEVMODE	Force Device Mode Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. ■ 1'b0: Normal Mode ■ 1'b1: Force Device Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 μ s is sufficient.	
[29]	R/W	FORCEHSTMODE	Force Host Mode Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. ■ 1'b0: Normal Mode ■ 1'b1: Force Host Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 μ s is sufficient.	
[27:16]	-	RESERVED	Reserved	
[15]	R/W	PHYLOWPOWER CLOCKSELECT	PHY Low-Power Clock Select Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY usually operate on a 48-MHz clock to save power. 0 : 480-MHz Internal PLL clock 1 : 48-MHz External clock *Note: This bit must be configured with OPHYPWRL.PLL_powerdown.	0x0
[14:10]	-	RESERVED	Reserved	0x5
[9]	R/W	HNPCAP	HNP - Capable The application uses this bit to control the OTG cores's HNP capabilities. 0 : HNP capability is not enabled	0x0

Bit	R/W	Symbol	Description	Reset Value
			1 : HNP capability is enabled	
[8]	R/W	SRPCAP	SRP - Capable The application uses this bit to control the OTG core's SRPcapabilities. 0 : SRP capability is not enabled 1 : SRP capability is enabled	0x0
[7:4]	-	RESERVED	Reserved	-
[3]	R/W	PHYIF	PHY Interface The application uses this bit to configure the core to support aUTMI+ PHY with an 8- or 16-bit interface. Only 16-bit interface is supported. This bit must be set to 1. 0 : 8 bits 1 : 16 bits	0x0
[2:0]	R/W	TOUTCAL	HS/ FS Timeout Calibration Set this bit to 3h7.	0x0
GRSTCTL Address : 0xC004_0010 Reset Value : 0x80000000				
[31]	R	AHBIDLE	AHB Master Idle Indicates that the AHB Master State Machine is in the IDLEcondition.	0x1
[30]	R	DMAREQ	DMA Request Signal Indicates that the DMA request is in progress. Used for debug.	0x0
[29:11]	-	RESERVED	Reserved	-
[10:6]	R/W	TXFNUM	TxFIFO Number This is the FIFO number. Use TxFIFO Flush bit to flush FIFOnumber. This field must not be changed until the core clears theTxFIFO Flush bit. 0 : Non-Periodic TxFIFO flush 1 : Periodic TxFIFO 1 flush in Device mode for Periodic TxFIFO flush in Host mode 2 : Periodic TxFIFO 2 flush in Device mode 15 : Periodic TxFIFO 15 flush in Device mode 16 : Flush all the Periodic and Non-Periodic TxFIFOs in the core	0x0
[5]	R/W	TXFFLSH	TxFIFO Flush This bit selectively flushes a single or all transmit FIFOs, butcannot flush if the core is in the middle of a transaction. Theapplication must only write this bit after checking that the core isneither writing to the TxFIFO nor reading from the TxFIFO. Theapplication must wait until the core clears this bit beforeperforming any operations. This bit takes 8 clocks to clear.	0x0
[4]	R/W	RXFFLSH	RxFIFO Flush The application flushes the entire RxFIFO using this bit, but mustfirst ensure that the core is not in the middle of a transaction. Theapplication must only write to this bit after checking that the coreis neither reading from the RxFIFO nor writing to the RxFIFO.The application must wait until the bit is cleared beforeperforming any other operations. This bit takes 8 clocks to clear.	0x0
[3]	R/W	INTKNQFLSH	IN Token Sequence Learning Queue Flush The application writes this bit to flush the IN Token SequenceLearning Queue.	0x0
[2]	R/W	FRMCNTRRST	Host Frame Counter Reset The application writes this bit to reset the (micro) frame numbercounter inside the core. If the (micro) frame counter is reset, thesubsequent SOF sent out by the core will have a (micro) framenumbers of 0.	0x0

Bit	R/W	Symbol	Description	Reset Value
[1]	R/W	HSFTRST	<p>HClk Soft Reset</p> <p>The application uses this bit to flush the control logic in the AHBClock domain. Only AHB Clock Domain pipelines are reset.</p> <p>FIFOs are not flushed with this bit.</p> <p>All state machines in the AHB clock Domain are reset to IDLEstate after terminating the transactions on the AHB, following the protocol.</p> <p>Control bits in the CSRs that the AHB Clock domain statemachines use are cleared.</p> <p>Status mask bits generated by the AHB Clock domain statemachine that control the interrupt status, are cleared to clear the interrupt.</p> <p>Because interrupt status bits are not cleared, the application gets the status of any core events that occurred after this bit isset.</p> <p>This is a self-clearing bit that the core clears after all necessarylogic is reset in the core. This may take several clocks,depending on the core's current state.</p>	0x0
[0]	R/W	CSFTRST	<p>Core Soft Reset</p> <p>Resets the hclk and phy_clock domains as follows:</p> <p>Clears the interrupts and all the CSR registers except thefollowing register bits:</p> <ul style="list-style-type: none"> - HCFG.FSLSPolkSel - DCFG.DevSpd <p>All module state machines (except the AHB Slave Unit) arerestet to the IDLE state, and all the transmit FIFOs and thereceive FIFO are flushed.</p> <p>Any transactions on the AHB Master are terminated as soon aspossible, after gracefully completing the last data phase of anAHB transfer. Any transactions on the USB are terminatedimmediately.</p> <p>The application can write to this bit any time it wants to reset thecore. This is a self-clearing bit and the core clears this bit after allthe necessary logic is reset in the core, which may take severalclocks, depending on the current state of the core. Once this bitis cleared software must wait at least 3 PHY clocks before doingany access to the PHY domain. Software must also check thatbit 31 of this register is 1 (AHB Master is IDLE) before startingany operation. Typically software reset is used during softwaredevelopment and if you dynamically change the PHY selectionbits in the USB configuration registers listed above. If youchange the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock isselected, the PHY domain has to be reset for proper operation.</p>	0x0
GINTSTS				
Address : 0xC004_0014				
Reset Value : 0x04001020				
[31]	R/W	WKUPINT	Resume/ Remote Wakeup Detected Interrupt In Device mode, this interrupt is asserted if a resume is detectedon the USB. In Host mode, this interrupt is asserted if a remotewakeup is detected on the USB.	0x0
[30]	R/W	SESSREQINT	Session Request/ New Session Detected Interrupt In Host mode, this interrupt is asserted if a session request isdetected from the device. In Device mode, this interrupt isasserted if the b_valid signal goes high.	0x0
[29]	R/W	DISCONNINT	Disconnect Detected Interrupt Asserted when a device disconnect is detected.	0x0
[28]	R/W	CONIDSTSCHNG	Connector ID Status Change The core sets this bit if there is a change in connector ID status.	0x0
[27]	-	RESERVED	Reserved	-
[26]	R	PTXFEMP	Periodic TxFIFO Empty Asserted if the Periodic Transmit FIFO is either half or completelyempty and there is space for at least one entry to be written in thePeriodic Request Queue. The half or completely empty status isdetermined by the Periodic TxFIFO Empty Level bit in the CoreAHB Configuration register.	0x0
[25]	R	HCHINT	Host Channels Interrupt	0x0

Bit	R/W	Symbol	Description	Reset Value
			The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-nInterrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.	
[24]	R	PRTINT	Host Port Interrupt The core sets this bit to indicate a change in port status of one of the OTG core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	0x0
[23]	-	RESERVED	Reserved	-
[22]	R/W	FETSUSP	Data Fetch Suspended. This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of Tx FIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: <ul style="list-style-type: none">■ Sets a global non-periodic IN NAK handshake■ Disables IN endpoints■ Flushes the FIFO■ Determines the token sequence from the IN Token Sequence Learning Queue■ Re-enables the endpoints■ Clears the global non-periodic IN NAK handshake If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token received by the core generates an "IN token received when FIFO empty" interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application checks the GINSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application masks the "IN token received when FIFO empty" interrupt if clearing a global IN NAK handshake.	0x0
[21]	R/W	INCOMPPIP	Incomplete Periodic Transfer. In Host mode, the core sets this interrupt bit if there are incomplete periodic transactions still pending which are scheduled for the current microframe.	0x0
		INCOMPLSOOUT	Incomplete Isochronous OUT Transfer. The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not complete in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	
[20]	R/W	INCOMPLETE	Isochronous IN Transfer. The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not complete in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	0x0
[19]	R	OEPINT	OUT Endpoints Interrupt. The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.	0x0
[18]	R	IEPINT	IN Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding	0x0

Bit	R/W	Symbol	Description	Reset Value
			Device IN Endpoint-n Interrupt (DIEPINTn) register to determinethe exact cause of the interrupt. The application must clear theappropriate status bit in the corresponding DIEPINTn register toclear this bit.	
[17]	R/W	EPMIS	Endpoint Mismatch Interrupt Indicates that an IN token has been received for a non-periodicendpoint, but the data for another endpoint is present in the topof the Non-Periodic Transmit FIFO and the IN endpointmismatch count programmed by the application has expired.	0x0
[16]	-	RESERVED	Reserved	-
[15]	R/W	EOPF	End of Periodic Frame Interrupt Indicates that the period specified in the Periodic Frame Intervalfield of the Device Configuration register (DCFG.PerFrInt) hasbeen reached in the current microframe.	0x0
[14]	R/W	ISOUTDROP	Isochronous OUT Packet Dropped Interrupt The core sets this bit if it fails to write an isochronous OUTpacket into the RxFIFO because the RxFIFO does not haveenough space to accommodate a maximum packet size packetfor the isochronous OUT endpoint.	0x0
[13]	R/W	ENUMDONE	Enumeration Done The core sets this bit to indicate that speed enumeration iscomplete. The application must read the Device Status (DSTS)register to obtain the enumerated speed.	0x0
[12]	R/W	USBRST	USB Reset The core sets this bit to indicate that a reset is detected on theUSB.	0x1
[11]	R/W	USBSUSP	USB Suspend The core sets this bit to indicate that a suspend was detected onthe USB.The core enters the Suspended state if there is noactivity on the line_state signal for an extended period of time.	0x0
[10]	R/W	ERLYSUSP	Early Suspend The core sets this bit to indicate that an Idle state has beendetected on the USB for 3 ms.	0x0
[9]	-	RESERVED	Reserved	-
[8]	-	RESERVED	Reserved	-
[7]	R	GOUTNAKEFF	Global OUT NAK Effective Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit is cleared by writing the Clear Global OUT NAK bit in the Device Control register.	0x0
[6]	R	GINNAKEFF	Global IN Non-Periodic NAK Effective Indicates that the Set Global Non-Periodic IN NAK bit in theDevice Control register (DCTL.SGNPIInNak), set by theapplication, has taken effect in the core. That is, the core hassampled the Global IN NAK bit set by the application. This bit iscleared by clearing the Clear Global Non-Periodic IN NAK bit setby the application. This bit is cleared by clearing the Clear Global Non-Periodic IN NAK bit in the Device Control register(DCTL.CGNPIInNak). This interrupt does not necessarily meanthat a NAK handshake is sent out on the USB. The STALL bittakes precedence over the NAK bit.	0x0
[5]	R	NPTXFEMP	Non-Periodic TxFIFO Empty This interrupt is asserted if the Non-Periodic TxFIFO is either halfor completely empty, and there is space for at least one entry tobe written to the Non-Periodic Transmit Request Queue. The halfor completely empty status is determined by the Non-PeriodicTxFIFO Empty Level bit in the Core AHB Configuration register(GAHBCFG.NPTxFEmpLvl).	0x1
[4]	R	RXFLVL	RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.	0x0

Bit	R/W	Symbol	Description	Reset Value
[3]	R/W	SOF	Start of (micro) Frame In Host mode, the core sets this bit to indicate that an SOF (FS),micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB.The application must write a 1 to this bit to clear the interrupt. InDevice mode, in the core sets this bit to indicate that an SOFToken has been received on the USB. The application reads theDevice Status register to get the current (micro) frame number.This interrupt is seen if the core is operating at either HS or FS.	0x0
[2]	R	OTGINT	OTG Interrupt The core sets this bit to indicate an OTG protocol event. Theapplication must read the OTG Interrupt Status (GOTGINT)register to determine the exact event that caused this interrupt.The application must clear the appropriate status bit in theGOTGINT register to clear this bit.	0x0
[1]	R/W	MODEMIS	Mode Mismatch Interrupt The core sets this bit if the application is trying to access: A Host mode register, if the core is operating in Device mode A Device mode register, if the core is operating in Host mode	0x0
[0]	R	CURMOD	Current Mode Of Operation Indicates the current mode of operation. 0 : Device mode 1 : Host mode	0x0
GINTMSK Address : 0xC004_0018 Reset Value : 0x00000000				
[31]	R/W	WKUPINTMSK	Resume/ Remote Wakeup Detected Interrupt Mask	0x0
[30]	R/W	SESSREQINTMSK	Session Request/ New Session Detected Interrupt Mask	0x0
[29]	R/W	DISCONNINTMSK	Disconnect Detected Interrupt Mask	0x0
[28]	R/W	CONIDSTSCHNGMSK	Connector ID Status Change Mask	0x0
[27]	-	RESERVED	Reserved	-
[26]	R/W	PTXFEMPMSK	Periodic TxFIFO Empty Mask	0x0
[25]	R/W	HCHINTMSK	Host Channels Interrupt Mask	0x0
[24]	R/W	PRTINTMSK	Host Port Interrupt Mask	0x0
[23]	-	RESERVED	Reserved	-
[22]	R/W	FETSSUSPMSK	Data Fetch Suspended Mask	0x0
[21]	R/W	INCOMPLPMSK	Incomplete Periodic Transfer Mask R/W	0x0
		INCOMPISOOUTMSK	Incomplete Isochronous OUT Transfer Mask	
[20]	R/W	INCOMPISOINMSK	Incomplete Isochronous IN Transfer Mask	0x0
[19]	R/W	OEPINTMSK	OUT Endpoints Interrupt Mask	0x0
[18]	R/W	INEPINTMSK	IN Endpoints Interrupt Mask	0x0
[17]	R/W	EPMISMSK	Endpoint Mismatch Interrupt Mask	0x0
[16]	-	RESERVED	Reserved	-
[15]	R/W	EOPFMSK	End of Periodic Frame Interrupt Mask	0x0
[14]	R/W	ISOOUTDROPMSK	Isochronous OUT Packet Dropped Interrupt Mask	0x0
[13]	R/W	ENUMDONEMSK	Enumeration Done Mask	0x0

Bit	R/W	Symbol	Description	Reset Value
[12]	R/W	USBRSTMSK	USB Reset Mask	0x0
[11]	R/W	USBSUSPMSK	USB Suspend Mask	0x0
[10]	R/W	ERLYSUSPMSK	Early Suspend Mask	0x0
[9]	-	RESERVED	Reserved	-
[8]	-	RESERVED	Reserved	-
[7]	R/W	GOUTNAKEFFMSK	Global OUT NAK Effective Mask	0x0
[6]	R/W	GINNAKEFFMSK	Global Non-Periodic IN NAK Effective Mask	0x0
[5]	R/W	NPTXFEMPMSK	Non-Periodic TxFIFO Empty Mask	0x0
[4]	R/W	RXFLVLMSK	Receive FIFO Non-Empty Mask	0x0
[3]	R/W	SOFMSK	Start of (micro)Frame Mask	0x0
[2]	R/W	OTGINTMSK	OTG Interrupt Mask	0x0
[1]	R/W	MODEMISMSK	Mode Mismatch Interrupt Mask	0x0
[0]	-	RESERVED	Reserved	-
GRXSTSR (Host Mode)				
Address : 0xC004_001C				
Reset Value :-				
[31:21]	-	RESERVED	Reserved	-
[20:17]	R	PKTSTS	Packet Status Indicates the status of the received packet. 2 : IN data packet received 3 : IN transfer completed (triggers an interrupt) 5 : Data toggle error (triggers an interrupt) 7 : Channel halted (triggers an interrupt) others : Reserved	-
[16:15]	R	DPID	Data PID Indicates the Data PID of the received packet. 0 : DATA0 2 : DATA1 1 : DATA2 3 : MDATA	-
[14:4]	R	BCNT	Byte Count Indicates the byte count of the received IN data packet.	-
[3:0]	R	CHNUM	Channel number Indicates the channel number to which the current received packet belongs.	-
GRXSTSP (Device Mode)				
Address : 0xC004_0020				
Reset Value : 0xFFFFFFFF				
[31:25]	-	RESERVED	Reserved	0x3F
[24:21]	R	FN	Frame Number This is the least significant 4 bits of the (micro) frame number inwhich the packet is received on the USB. This field is supported if isochronous OUT endpoints are supported.	0xF

Bit	R/W	Symbol	Description	Reset Value
[20:17]	R	PKTSTS	Packet Status Indicates the status of the received packet. 1 : Global OUT NAK (triggers an interrupt) 2 : OUT data packet received 3 : OUT transfer completed (triggers an interrupt) 4 : SETUP transaction completed (triggers an interrupt) 6 : SETUP data packet received others: Reserved	0xF
[16:15]	R	DPID	Data PID Indicates the Data PID of the received OUT data packet. 0 : DATA0 2 : DATA1 1 : DATA2 3 : MDATA	0x3
[14:4]	R	BCNT	Byte Count Indicates the byte count of the received data packet.	0x3FF
[3:0]	R	EPNUM	Endpoint number Indicates the endpoint number to which the current received packet belongs.	0xF
GRXFSIZ				
Address : 0xC004_0024				
Reset Value : 0x00001800				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	RXFDEP	RxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 6144 The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. A new value must be written to this field. Programmed values must not exceed the power-on value set.	0x1800
GNPTXFSIZ				
Address : 0xC004_0028				
Reset Value : 0x18001800				
[31:16]	R/W	NPTXFDEP	Non-Periodic TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32768 The power-on reset value of this register is specified as the Largest Non-Periodic Tx Data FIFO Depth (6144). A new value must be written to this field. Programmed values must not exceed the power-on value set.	0x1800
[15:0]	R/W	NPTXFSTADDR	Non-Periodic Transmit Start Address This field contains the memory start address for Non-Periodic Transmit FIFO RAM. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth (6144). A new value must be written to this field. Programmed values must not exceed the power-on value set.	0x1800

Bit	R/W	Symbol	Description	Reset Value
GNPTXSTS Address : 0xC004_002C Reset Value : 0x00081800				
[31]	-	RESERVED	Reserved	-
[30:24]	R	NPTXQTOP	<p>Top of the Non-Periodic Transmit Request Queue.</p> <p>Entry in the Non-Periodic Tx Request Queue that is currently being processed by the MAC.</p> <p>Bits[30:27] : Channel/ endpoint number</p> <p>Bits[26:25] :</p> <p>0 : IN/ OUT token</p> <p>1 : Zero-length transmit packet (device IN/host OUT)</p> <p>2 : PING/CSPLIT token</p> <p>3 : Channel halt command</p> <p>Bit[24] : Terminate (last entry for selected channel/endpoint)</p>	0x0
[23:16]	R	NPTXQSPCAVAIL	<p>Non-Periodic Transmit Request Queue Space Available.</p> <p>Indicates the amount of free space available in the Non-Periodic Transmit Request Queue. This queue holds both INand OUT requests in Host mode. Device mode has only INrequests.</p> <p>8'h0 : Non-Periodic Transmit Request Queue is full</p> <p>8'h1 : 1 location available</p> <p>8'h2 : 2 locations available</p> <p>n : n locations available(0≤n≤8)</p> <p>Others : Reserved</p>	0x8
[15:0]	R	NPTXFSPCAVAIL	<p>Non-Periodic TxFIFO Space Available</p> <p>Indicates the amount of free space available in the Non-Periodic TxFIFO.</p> <p>Values are in terms of 32-bit words.</p> <p>0 : Non-Periodic TxFIFO is full</p> <p>1 : 1 word available</p> <p>2 : 2 words available</p> <p>n : n words available (where 0 ≤ n ≤ 32768)</p> <p>0x8000 : 32768 words available</p> <p>Others : Reserved</p>	0x1800
HPTXFSIZ Address : 0xC004_0100 Reset Value : 0x03005A00				
[31:16]	R/W	PTXFSIZE	<p>Host Periodic TxFIFO Depth</p> <p>This value is in terms of 32-bit words</p> <p>Minimum value is 16</p> <p>Maximum value is 6144</p> <p>A new value must be written to this field. Programmed valuesmust not exceed the Maximum value.</p>	0x0300
[15:0]	R/W	PTXFSTADDR	<p>Host Periodic TxFIFO Start Address</p> <p>The power-on reset value of this register is sum of theLargest Rx Data FIFO Depth and Largest Non -Periodic TxDATA FIFO Depth specified.</p> <p>If you have programmed new values for the RxFIFO or Non-Periodic TxFIFO, write their sum in this field. Programmedvalues must not exceed the power-on value.</p>	0x5A00
DPTXFSIZn (DPTXFSIZ1, DPTXFSIZ2, ..., DPTXFSIZ15) Address : 0xC004_0104 + (n-1)*0x4 Reset Value : 0x03001000, ..., 0x03004800				

Bit	R/W	Symbol	Description	Reset Value
[31:16]	R/W	DPTXFSIZE	<p>Device Periodic TxFIFO Size This value is in terms of 32-bit words Minimum value is 4 Maximum value is 768 The power-on reset value of this register is the LargestDevice Mode Periodic Tx Data FIFO Depth. Write a newvalue to this field.</p>	n:1 (0x0300) n:2 (0x0300) n:3 (0x0300) n:4 (0x0300) n:5 (0x0300) n:6 (0x0300) n:7 (0x0300) n:8 (0x0300) n:9 (0x0300) n:10(0x0300) n:11(0x0300) n:12(0x0300) n:13(0x0300) n:14(0x0300) n:15(0x0300)
[15:0]	R/W	DPTXFSTADDR	<p>Device Periodic TxFIFO RAM Start AddressHolds the start address in the RAM for this periodic FIFO. The power-on reset value of this register is sum of theLargest Rx Data FIFO Depth, Largest Non-Periodic Tx DataFIFO Depth, and all lower numbered Largest Device ModePeriodic Tx Data FIFOOn Depth specified. If you have programmed new values for the RxFIFO, Non-Periodic TxFIFO, or device Periodic TxFIFOs, write theirsum in this field. Programmed values must not exceed thepower-on value set.</p>	n:1 (0x1000) n:2 (0x3300) n:3 (0x3600) n:4 (0x3900) n:5 (0x3C00) n:6 (0x3F00) n:7 (0x4200) n:8 (0x4500) n:9 (0x4800) n:10(0x4B00) n:11(0x4E00) n:12(0x5100) n:13(0x5400) n:14(0x5700) n:15(0x5A00)

25.4.4 Host Mode Registers (USB_OTG_HMCSR)

Bit	R/W	Symbol	Description	Reset Value
HCFG				
Address : 0xC004_0400				
Reset Value : 0x00200000				
[31:3]	-	RESERVED	Reserved	0x0040000
[2]	R/W	FSLSSUPP	<p>FS- and LS- Only Support The application uses this bit to control the core's enumerationspeed. Using this bit, the application makes the core enumerateas a FS host, even if the connected device supports HS traffic.Do not make changes to this field after initial programming.</p> <p>0 : HS/FS/LS, based on the maximum speed supported by the connected device 1 : FS/LS-only, even if the connected device can support HS</p>	0x0
[1:0]	R/W	FSLSPCLKSEL	FS/ LS PHY Clock Select If the core is in FS Host mode	-

Bit	R/W	Symbol	Description	Reset Value
			0 : PHY clock is 30/60 MHz 1 : PHY clock is 48 MHz Others : Reserved If the core is in LS Host mode 0 : PHY clock is 30/60 MHz 1 : PHY clock is 48 MHz 2 : PHY clock is 6 MHz 3 : Reserved	
HFIR <i>Address : 0xC004_0404</i> <i>Reset Value : 0x000017D7</i>				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	FRINT	Frame Interval The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHYclocks that constitute the required frame interval. The default value set in this field for a FS operation if the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHYclock specified in the FS/LS PHY Clock Select field of the HostConfiguration register (HCFG.FSLSPdckSel). Do not change the value of this field after the initial configuration. 125 µs * (PHY clock frequency for HS) 1 ms * (PHY clock frequency for FS/LS)	0x17D7
HNUM <i>Address : 0xC004_0408</i> <i>Reset Value : 0x00000000</i>				
[31:16]	R	FRREM	Frame Time Remaining Indicates the amount of time remaining in the current microframe(HS) or frame (FS/ LS), in terms of PHY clocks. This field decrements on each PHY clock. If it reaches zero, this field is reloaded with the value in the Frame Interval register and a newSOF is transmitted on the USB.	0x0
[15:0]	R	FRNUM	Frame Number This field increments if a new SOF is transmitted on the USB, and is reset to 0 if it reaches 0x3FFF.	0x0
HPTXSTS <i>Address : 0xC004_0410</i> <i>Reset Value : 0x00080100</i>				
[31:24]	R	PTXQTOP	Top of the Periodic Transmit Request Queue This indicates the entry in the Periodic Tx Request Queue that is currently being processed by the MAC. This register is used for debugging. Bit [31] : Odd/Even (micro)frame - 0 : send in even (micro)frame - 1 : send in odd (micro)frame Bits [30:27] : Channel/endpoint number Bits [26:25] : Type - 0 : IN/OUT	0x0

Bit	R/W	Symbol	Description	Reset Value
			- 1 : Zero-length packet - 2 : CSPLIT - 3 : Disable channel command Bit[24] : Terminate	
[23:16]	R	PTXQSPCAVAIL	Periodic Transmit Request Queue Space Available Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. 0 : Periodic Transmit Request Queue is full 1 : 1 location available 2 : 2 location available n : n locations available (0 ≤ n ≤ 8) Others : Reserved	0x8
[15:0]	R	PTXFSPCAVAIL	Periodic Transmit Data FIFO Space Available Indicates the number of free locations available to be written to the Periodic TxFIFO. Values are in terms of 32-bit words 0: Periodic TxFIFO is full 1: 1 word available 2: 2 words available n: n words available (0 ≤ n ≤ 8) Others: Reserved	0x0100
HAINT Address : 0xC004_0414 Reset Value : 0x00000000				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R	HAINT	Channel Interrupts One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	0x0
HAINTMSK Address : 0xC004_0418 Reset Value : 0x00000000				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	HAINTMSK	Channel Interrupt Mask One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	0x0
HPRT Address : 0xC004_0440 Reset Value : 0x00000000				
[31:19]	-	RESERVED	Reserved	-
[18:17]	R	PRTSPD	Port Speed Indicates the speed of the device attached to this port. 0 : High speed 1 : Full speed 2 : Low speed 3 : Reserved	0x0
[16:13]	R/W	PRTTSTCTL	Port Test Control The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port.	0x0

Bit	R/W	Symbol	Description	Reset Value
			0 : Test mode disabled 1 : Test_J mode 2 : Test_K mode 3 : Test_SE0_NAK mode 4 : Test_Packet mode 5 : Test_Force_Enable Others : Reserved	
[12]	R/W	PRTPWR	Port Power The application uses this field to control power to this port, and the core clears this bit on an overcurrent condition. 0 : Power off 1 : Power on	0x0
[11:10]	R	PRTLNSTS	Port Line Status Indicates the current logic level USB data lines Bit [10] : Logic level of D? Bit [11] : Logic level of D+	0x0
[9]	-	RESERVED	Reserved	-
[8]	R/W	PTRTRST	Port Reset If the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. 0 : Port not in reset 1 : Port in reset The application must leave this bit set for at least a minimum duration mentioned below to start a reset on the port. The application can leave it set for another 10ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard. High speed : 50 ms Full speed/Low speed : 10ms	0x0
[7]	R/W	PRTSUSP	Port Suspend The application sets this bit to put this port in Suspend mode. The core stops sending SOFs if this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register. 0 : Port not in Suspend mode 1 : Port in Suspend mode	0x0
[6]	R/W	PRTRES	Port Resume The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register, the core starts driving resume signaling without application intervention and clears this bit if it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. 0 : No resume driven 1 : Resume driven	0x0
[5]	R/W	PTOVRCURRCHNG	Port Overcurrent Change The core sets this bit if the status of the Port Overcurrent Active bit (bit 4) in this register	0x0

Bit	R/W	Symbol	Description	Reset Value
			changes.	
[4]	R	PRTOVRCURRACT	Port Overcurrent Active Indicates the overcurrent condition of the port. 0 : No overcurrent condition 1 : Overcurrent condition	0x0
[3]	R/W	PRTENCHNG	Port Enable/Disable Change The core sets this bit if the status of the Port Enable bit [2] of this register changes.	0x0
[2]	R/W	PRTENA	Port Enable A port is enabled by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit. The application cannot set this bit by a register write. It clears it to disable the port. This bit does not trigger any interrupt to the application. 0 : Port disabled 1 : Port enabled	0x0
[1]	R/W	PRTCNNDDET	Port Connect Detected The core sets this bit if a device connection is detected to trigger an interrupt to the application using the Host Port interrupt bit of the Core Interrupt register. The application must write a 1 to this bit to clear the interrupt.	0x0
[0]	R	PRTCONNSTS	Port Connect Status 0 : No device is attached to the port 1 : A device is attached to the port	0x0
HCCHARn (0 ≤ n ≤ 15, HCCHAR0, HCCHAR1, ..., HCCHAR15) Address : 0xC004_0500 + n*0x20 Reset Value : 0x00000000				
[31]	R/W	CHENA	Channel Enable This field is set by the application and cleared by the OTG host. 0 : Disables Channel 1 : Enables Channel	0x0
[30]	R/W	CHDIS	Channel Disable The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.	0x0
[29]	R/W	ODDFRM	Odd Frame This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro) frame. This field is applicable for only periodic transactions. 0 : Even (micro)frame 1 : Odd (micro)frame	0x0
[28:22]	R/W	DEVADDR	Device Address This field selects the specific device serving as the data source or sink.	0x0
[21:20]	R/W	MC/EC	Multi Count/Error Count If the Split Enable bit of the Host Channel-n Split Control register is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this endpoint. 0 : Reserved 1 : 1 transaction 2 : 2 transactions to be issued for this endpoint per microframe 3 : 3 transactions to be issued for this endpoint per microframe	0x0

Bit	R/W	Symbol	Description	Reset Value
			If HCSPLTr.SplEna is set, this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 0x1.	
[19:18]	R/W	EPTYPE	Endpoint Type Indicates the transfer type selected. 0 : Control 1 : Isochronous 2 : Bulk 3 : Interrupt	0x0
[17]	R/W	LSPDDEV	Low-Speed Device This field is set by the application to indicate that this channel is communicating to a low-speed device.	0x0
[16]	-	RESERVED	Reserved	-
[15]	R/W	EPDIR	Endpoint Direction Endpoint Type Indicates the transfer type selected. 0 : OUT 1 : IN	0x0
[14:11]	R/W	EPNUM	Endpoint Number Indicates the endpoint number on the device serving as the datasource or sink.	0x0
[10:0]	R/W	MPS	Maximum Packet Size Indicates the maximum packet size of the associated endpoint.	0x0
HCSPLTn (0 ≤ n ≤ 15, HCSPLT0, HCSPLT1, ..., HCSPLT15)				
Address : 0xC004_0504 + n*0x20				
Reset Value : 0x00000000				
[31]	R/W	SPLTENA	Split Enable The application sets this field to indicate that this channel is enabled to perform split transactions.	0x0
[30:17]	-	RESERVED	Reserved	-
[16]	R/W	COMPSPLT	Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.	0x0
[15:14]	R/W	XACTPOS	Transaction Position This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. 3: All. This is the entire data payload of this transaction. 2: Begin. This is the first data payload of this transaction. 0: Mid. This is the middle payload of this transaction. 1: End. This is the last payload of this transaction.	0x0
[13:7]	R/W	HUBADDR	Hub Address This field holds the device address of the transaction translator's hub.	0x0
[6:0]	R/W	PRTADDR	Port Address This field is the port number of the recipient transaction translator.	0x0
HCINTn (0 ≤ n ≤ 15, HCINT0, HCINT1, ..., HCINT15)				
Address : 0xC004_0508 + n*0x20				
Reset Value : 0x00000000				

Bit	R/W	Symbol	Description	Reset Value
[31:11]	-	RESERVED	Reserved	-
[10]	R/W	DATATGLERR	Data Toggle Error	0x0
[9]	R/W	FRMOVRUN	Frame Overrun	0x0
[8]	R/W	BBLERR	Babble Error	0x0
[7]	R/W	XACTERR	Transaction Error	0x0
[6]	R/W	NYET	NYET Response Received Interrupt	0x0
[5]	R/W	ACK	ACK Response Received Interrupt	0x0
[4]	R/W	NAK	NAK Response Received Interrupt	0x0
[3]	R/W	STALL	STALL Response Received Interrupt	0x0
[2]	R/W	AHBERR	AHB Error This is generated only in Internal DMA mode if there is an AHB error during AHB read/writes. The application reads the corresponding channel's DMA address register to get the error address.	0x0
[1]	R/W	CHHLTD	Channel Halted Indicates the incomplete transfer either because of any USB transaction error or in response to disable request by the application.	0x0
[0]	R/W	XFERCOMPL	Transfer Completed Transfer completed normally without any errors.	0x0

HCINTMSKn ($0 \leq n \leq 15$, HCINTMSK0, HCINTMSK1, ..., HCINTMSK15)

Address : 0xC004_050C + n*0x20

Reset Value : 0x00000000

[31:11]	-	RESERVED	Reserved	-
[10]	R/W	DATATGLERRMSK	Data Toggle Error Mask	0x0
[9]	R/W	FRMOVRUNMSK	Frame Overrun Mask	0x0
[8]	R/W	BBLERRMSK	Babble Error Mask	0x0
[7]	R/W	XACTERRMSK	Transaction Error Mask	0x0
[6]	R/W	NYETMSK	NYET Response Received Interrupt Mask	0x0
[5]	R/W	ACKMSK	ACK Response Received Interrupt Mask	0x0
[4]	R/W	NAKMSK	NAK Response Received Interrupt Mask	0x0
[3]	R/W	STALLMSK	STALL Response Received Interrupt Mask	0x0
[2]	R/W	AHBERRMSK	AHB Error Mask	0x0
[1]	R/W	CHHLTDMSK	Channel Halted Mask	0x0
[0]	R/W	XFERCOMPLMSK	Transfer Completed Mask	0x0

HCTSIZn ($0 \leq n \leq 15$, HCTSIZ0, HCTSIZ1, ..., HCTSIZ15)

Address : 0xC004_0510 + n*0x20

Reset Value : 0x00000000

[31]	R/W	DOPNG	Do Ping Setting this field to 1 directs the host to do PING protocol.	0x0
[30:29]	R/W	PID	PID The application programs this field with the type of PID to use for the initial transaction.	0x0

Bit	R/W	Symbol	Description	Reset Value
			The host maintains this field for therest of the transfer. 0: DATA0 1: DATA1 2: DATA2 3: MDATA (non-control)/ SETUP(control)	
[28:19]	R/W	PKTCNT	Packet Count This field is programmed by the application with the expectednumber of packets to be transmitted (OUT) or received (IN).The host decrements this count on every successfultransmission or reception of an OUT/ IN packet. Once thiscount reaches zero, the application is interrupted to indicate normal completion.	0x0
[18:0]	R/W	XFERSIZE	Transfer Size For an OUT, this field is the number of data bytes the hostsends during the transfer.For an IN, this field is the buffer size that the application has reserved for the transfer. The application is expected toprogram this field as an integer multiple of the maximumpacket size for IN transactions.	0x0
HCDMA_n (0 ≤ n ≤ 15, HCDMA0, HCDMA1,..., HCDMA15) Address : 0xC004_0514 + n*0x20 Reset Value : 0x00000000				
[31:0]	R/W	DMAADDR	DMA Address This field holds the start address in the external memory fromwhich the data for the endpoint must be fetched or to which it mustbe stored. This register is incremented on every AHB transaction.	0x0

25.4.5 Device Mode Regisers (USB_OTG_DMCSR)

Bit	R/W	Symbol	Description	Reset Value
DCFG Address : 0xC004_0800 Reset Value : 0x00200000				
[31:26]	R/W	RESVALID	Resume Validation Period This field controls the period when the core resumes from a suspend. When this bit is set, the core counts for the ResValid number of clock cycles to detect a valid resume. This field is effective only when DCFG.Ena32KHzSusp is set.	0x2
[25:24]	R/W	PERSCHINTVL	Periodic Scheduling Interval PerSchIntvl must be programmed only for Scatter/Gather DMA mode. Description: This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame. <ul style="list-style-type: none"> ■ When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. ■ When no periodic endpoints are active, then the internal DMA engine services nonperiodic endpoints, ignoring this field. ■ After the specified time within a (micro) frame, the DMA switches to fetching for nonperiodic endpoints. ■ 2b00: 25% of (micro) frame. ■ 2b01: 50% of (micro) frame. ■ 2b10: 75% of (micro) frame. ■ 2b11: Reserved. 	0x0

Bit	R/W	Symbol	Description	Reset Value
[23]	R/W	DESCDMA	<p>Enable Scatter/Gather DMA in Device mode</p> <p>When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation.</p> <p>NOTE: This bit must be modified only once after a reset.</p> <p>The following combinations are available for programming:</p> <ul style="list-style-type: none"> ■ GAHBCFG.DMAEn=0, DCFG.DescDMA=0 => Slave mode ■ GAHBCFG.DMAEn=0, DCFG.DescDMA=1 => Invalid ■ GAHBCFG.DMAEn=1, DCFG.DescDMA=0 => Buffered DMA mode ■ GAHBCFG.DMAEn=1, DCFG.DescDMA=1 => Scatter/Gather DMA mode 	0x0
[22:18]	R/W	EPMISCNT	<p>IN Endpoint Mismatch Count</p> <p>The application programs this field with a count that determines when the core generates an Endpoint Mismatch interrupt. The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or if the counter expires. The width of this counter depends on the depth of the Token Queue.</p>	0x8
[17:13]	-	RESERVED	Reserved	-
[12:11]	R/W	PERFRINT	<p>Periodic Frame Interval</p> <p>Indicates the time within a (micro) frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete.</p> <p>0 : 80% of the (micro) frame interval 1 : 85% 2 : 90% 3 : 95%</p>	0x0
[10:4]	R/W	DEVADDR	<p>Device Address</p> <p>The application must program this field after every SetAddress control command.</p>	0x0
[3]	R/W	ENA32KHZS	<p>Enable 32-KHz Suspend Mode</p> <p>When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend. This bit can only be set if USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero.</p>	0x0
[2]	R/W	NZSTSOUTSHHK	<p>Non-Zero-Length Status OUT Handshake</p> <p>The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage.</p> <ul style="list-style-type: none"> ■ 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. ■ 1'b0: Send the received OUT packet to the application (zero-length or nonzero length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register. 	0x0
[1:0]	R/W	DEVSPD	<p>Device Speed.</p> <p>Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected.</p> <ul style="list-style-type: none"> ■ 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ■ 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ■ 2'b10: Low speed (USB 1.1 FS transceiver clock is 48 MHz) ■ 2'b11: Full speed (USB 1.1 FS transceiver clock is 48 MHz) 	0x0

Bit	R/W	Symbol	Description	Reset Value
DCTL Address : 0xC004_0804 Reset Value : 0x00000000				
[31:17]	-	RESERVED	Reserved	-
[16]	R/W	NAKONBBLE	Set NAK automatically on babble The core sets NAK automatically for the endpoint on which babble is received.	0x0
[15]	R/W	IGNRFRMNUM	Ignore frame number for isochronous endpoints <ul style="list-style-type: none"> ■ Slave Mode (GAHBCFG.DMAEn=0): This bit is not valid in Slave mode and should not be programmed to 1. ■ Non-Scatter/Gather DMA mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=0): This bit is not used when Threshold mode is enabled and should not be programmed to 1. In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro) frames are completed. When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro) frames. <ul style="list-style-type: none"> - 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame - 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. This field is also used by the application to enable periodic transfer interrupts. ■ Scatter/Gather DMA Mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=1): This bit is not applicable to high-speed, high-bandwidth transfers and should not be programmed to 1. In addition, this bit is not used when Threshold mode is enabled and should not be programmed to 1. <ul style="list-style-type: none"> - 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. - 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. When this bit is set, there must be only one packet per descriptor. 	0x0
[14:13]	R/W	GMC	Global Multi Count GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for nonperiodic end points. <ul style="list-style-type: none"> ■ 2'b00: Invalid. ■ 2'b01: 1 packet. ■ 2'b10: 2 packets. ■ 2'b11: 3 packets. The value of this field automatically changes to 2'h1 when DCFG.DescDMA is set to 1. When Scatter/Gather DMA mode is disabled, this field is reserved and reads 2'b00.	0x0
[12]	-	RESERVED	Reserved	-
[11]	R/W	PWRONPRGDONE	Power-On Programming Done The application uses this bit to indicate that register programming is complete after a wake-up from Power Downmode.	0x0
[10]	W	CGOUTNAK	Clear Global OUT NAK	0x0

Bit	R/W	Symbol	Description	Reset Value
			A write to this field clears the Global OUT NAK.	
[9]	W	SGOUTNAK	Set Global OUT NAK A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set this bit after making sure that the Global OUT NAK Effective bit in Core Interrupt Register is cleared.	0x0
[8]	W	CGNPINNAK	Clear Global Non-Periodic IN NAK A write to this field clears the Global Non-Periodic IN NAK.	0x0
[7]	W	SGNPINNAK	Set Global Non-Periodic IN NAK A write to this field sets the Global Non-Periodic IN NAK. The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core sets this bit if a timeout condition is detected on a non-periodic endpoint. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register is cleared.	0x0
[6:4]	R/W	TSTCTL	Test Control 0 : Test mode disabled 1 : Test_J mode 2 : Test_K mode 3 : Test_SE0_NAK mode 4 : Test_Packet mode 5 : Test_Force_Enable Others : Reserved	0x0
[3]	R	GOUTNAKSTS	Global OUT NAK Status 0 : A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 1 : No data is written to the Rx FIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.	0x0
[2]	R	GNPINNAKSTS	Global Non-Periodic IN NAK Status 0 : A handshake is sent based on the data availability in the transmit FIFO. 1 : A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.	0x0
[1]	R/W	SFTDISCON	Soft Disconnect The application uses this bit to signal the OTG core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device will not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. 0 : Normal operation. If this bit is cleared after a soft disconnect, the core drives the opmode signal on the UTMI+ to 0x0, which generates a device connect event to the USB host. If the device is reconnected, the USB host restarts device enumeration. 1 : The core drives the opmode signal on the UTMI+ to 0x1, which generates a device disconnect event to the USB host.	0x0
[0]	R/W	RMTWKUPSIG	Remote Wakeup Signaling If the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15ms after setting it.	0x0
DSTS Address : 0xC004_0808				

Bit	R/W	Symbol	Description	Reset Value
Reset Value : 0x00000002				
[31:22]	-	RESERVED	Reserved	-
[21:8]	R	SOFFN	Frame or Microframe Number of the Received SOF If the core is operating at high speed; this field contains amicroframe number. If the core is operating at full or low speed, this field contains a frame number.	0x0
[7:4]	-	RESERVED	Reserved	-
[3]	R	ERRTICERR	The core sets this bit to report any erratic errors seen on theUTMI+. Due to erratic errors, the OTG core goes into Suspendedstate and an interrupt is generated to the application with EarlySuspend bit of the Core Interrupt register. If the early suspend is asserted due to an erratic error, the application performs a soft disconnect recover.	0x0
[2:1]	R	ENUMSPD	Enumerated Speed Indicates the speed at which the OTG core has come up afterspeed detection through a chirp sequence. 0: High speed (PHY clock is 30 MHz or 60 MHz) 1: Full speed (PHY clock is 30 MHz or 60 MHz) 2: Low speed (PHY clock is 6 MHz). 3: Full speed (PHY clock is 48 MHz). Low speed is not supported for devices using a UTMI+ PHY.	0x1
[0]	R	SUSPSTS	Suspend Status In device mode, this bit is set as long as a Suspend condition isdetected on the USB. The core enters the Suspended state ifthere is no activity on the line_state signal for an extended periodof time. The core comes out of the suspend:If there is any activity on the line_state signallf the application writes to the Remote Wakeup Signaling bit inthe Device Control register.	0x0
DIEPMSK Address : 0xC004_0810 Reset Value : 0x00000000				
[31:14]	-	RESERVED	Reserved	-
[13]	R/W	NAKMSK	NAK interrupt Mask	0x0
[12:10]	-	RESERVED	Reserved	-
[9]	R/W	BNAINTRMSK	BNA Interrupt Mask This bit is valid only when Device Descriptor DMA is enabled.	0x0
[8]	R/W	TXFIFOUDRNMSK	Fifo Underrun Mask	0x0
[7]	-	RESERVED	Reserved	-
[6]	R/W	INEPNAKEFFMSK	IN Endpoint NAK Effective Mask	0x0
[5]	R/W	INTKNEPMISMSK	IN Token received with EP Mismatch Mask	0x0
[4]	R/W	INTKNTXFEMPMSK	IN Token received with TxFIFO Empty mask	0x0
[3]	R/W	TIMEOUTMSK	Timeout Condition Mask	0x0
[2]	R/W	AHBERRMSK	AHB Error Mask	0x0
[1]	R/W	EPDISBLDMSK	Endpoint Disabled Interrupt Mask	0x0
[0]	R/W	XFERCOMPLMSK	Transfer Completed Interrupt Mask	0x0

Bit	R/W	Symbol	Description	Reset Value
DOEPMSK Address : 0xC004_0814 Reset Value : 0x00000000				
[31:15]	-	RESERVED	Reserved	-
[14]	R/W	NYETMSK	NYET Interrupt Mask	0x0
[13]	R/W	NAKMSK	NAK Interrupt Mask	0x0
[12]	R/W	BBLEERRMSK	Babble Interrupt Mask	0x0
[11:10]	-	RESERVED	Reserved	-
[9]	R/W	BNAOUTINTRMSK	BNA interrupt Mask	0x0
[8]	R/W	OUTPKTERRMSK	OUT Packet Error Mask	0x0
[7]	-	RESERVED	Reserved	-
[6]	R/W	BACK2BACKSETUP	Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.	0x0
[5]	-	RESERVED	Reserved	-
[4]	R/W	OUTTKNEPDISMSK	OUT Token Received When Endpoint Disabled Applies to control OUT endpoints only.	0x0
[3]	R/W	SETUPMSK	SETUP Phase Done Mask Applies to control endpoints only.	0x0
[2]	R/W	AHBERRMSK	AHB Error	0x0
[1]	R/W	EPDISBLDMSK	Endpoint Disabled Interrupt Mask	0x0
[0]	R/W	XFERCOMPLMSK	Transfer Completed Interrupt Mask	0x0
DAINT Address : 0xC004_0818 Reset Value : 0x00000000				
[31:16]	R	OUTEPINT	OUT Endpoint Interrupt Bits One bit per OUT endpoint : Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	0x0
[15:0]	R	INEPINT	IN Endpoint Interrupt Bits One bit per IN endpoint : Bit 0 for IN endpoint 0, bit 15 for endpoint 15	0x0
DAINTMSK Address : 0xC004_081C Reset Value : 0x00000000				
[31:16]	R/W	OUTEPMSK	OUT EP Interrupt Mask Bits One bit per OUT endpoint : Bit 16 for OUT EP 0, bit 31 for OUT EP 15	0x0
[15:0]	R/W	INEPMSK	IN EP Interrupt Mask Bits One bit per IN endpoint : Bit 0 for IN EP 0, bit 15 for IN EP 15	0x0
DTKNQR1 Address : 0xC004_0820 Reset Value : 0x00000000				

Bit	R/W	Symbol	Description	Reset Value
[31:8]	R	EPTKN	Endpoint Token Four bits per token represent the endpoint number of the token : Bits [31:28] : Endpoint number of Token 5 Bits [27:24] : Endpoint number of Token 4 ... Bits [15:12] : Endpoint number of Token 1 Bits [11:8] : Endpoint number of Token 0	0x0
[7]	R	WRAPBIT	Wrap Bit This bit is set if the write pointer wraps. It is cleared if the learning queue is cleared.	0x0
[6:5]	-	RESERVED	Reserved	-
[4:0]	R	INTKNWPTR	IN Token QUEUE Write Pointer	0x0
DTKNQR2 Address : 0xC004_0824 Reset Value : 0x00000000				
[31:0]	R	EPTKN	Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28] : Endpoint number of Token 13 Bits [27:24] : Endpoint number of Token 12 ... Bits [7:4] : Endpoint number of Token 7 Bits [3:0] : Endpoint number of Token 6	0x0
DVBUSDIS Address : 0xC004_0828 Reset Value : 0x000017D7				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	DVBUSDIS	Device VBUS Discharge Time Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals : VBUS discharge time in PHY clocks /1,024	0x17D7
DVBUSPULSE Address : 0xC004_082C Reset Value : 0x000005B8				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	DVBUSPULSE	Device VBUS Pulsing Time Specifies the VBUS pulsing time during SRP. This value equals : VBUS pulse time in PHY clocks /1,024	0x5B8
DTKNQR3 Address : 0xC004_0830 Reset Value : 0x00000000				
[31:0]	R	EPTKN	Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 21 Bits [27:24] : Endpoint number of Token 20 ...	0x0

Bit	R/W	Symbol	Description	Reset Value
			Bits [7:4] : Endpoint number of Token 15 Bits [3:0] : Endpoint number of Token 14	
DTKNQR4 <i>Address : 0xC004_0834</i> <i>Reset Value : 0x00000000</i>				
[31:0]	R	EPTKN	Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28] : Endpoint number of Token 29 Bits [27:24] : Endpoint number of Token 28 ... Bits [7:4] : Endpoint number of Token 23 Bits [3:0] : Endpoint number of Token 22	0x0
DIEPCTL0 <i>Address : 0xC004_0900</i> <i>Reset Value : 0x00008000</i>				
[31]	R/W	EPENA	Endpoint Enable Indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting any of the following interrupts on this endpoint. Endpoint Disabled Transfer Completed	0x0
[30]	R/W	EPDIS	Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	0x0
[29:28]	-	RESERVED	Reserved	-
[27]	W	SETNAK	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core also sets this bit for an endpoint after a SETUP packet is received on that endpoint.	0x0
[26]	W	CNAK	Clear NAK A write to this bit clears the NAK bit for the endpoint.	0x0
[25:22]	R	TXFNUM	TxFIFO Number This value is always set to 0, indicating that control IN endpoint0 data is always written in the Non-Periodic Transmit FIFO.	0x0
[21]	R/W	STALL	STALL Handshake The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.	0x0
[20]	-	RESERVED	Reserved	-
[19:18]	R	EPTYPE	Endpoint Type Hardcoded to 00 for control	0x0
[17]	R	NAKSTS	NAK Status Indicates the following:	0x0

Bit	R/W	Symbol	Description	Reset Value
			0 : The core is transmitting non-NAK handshakes based on the FIFO status 1 : The core is transmitting NAK handshakes on this endpoint If this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	
[16]	-	RESERVED	Reserved	-
[15]	R	USBACTEP	USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.	0x1
[14:11]	R/W	NEXTEP	Next Endpoint Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core accesses this field, even if the Endpoint Enable bit is not set. This field is not valid in Slave mode operation.	0x0
[10:2]	-	RESERVED	Reserved	-
[1:0]	R/W	MPS	Maximum Packet Size The application must program this field with the maximum packet size for the current logical endpoint. 0 : 64 bytes 1 : 32 bytes 2 : 16 bytes 3 : 8 bytes	0x0
DIEPINT0 Address : 0xC004_0908 Reset Value : 0x00000000				
[31:7]	-	EPENA	Reserved	-
[6]	R	INEPNAKEFF	IN Endpoint NAK Effective Applies to periodic IN endpoints only. Indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This bit is cleared if the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set. This interrupt does not necessarily mean that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.	0x0
[5]	R/W	INTKNEPMIS	IN Token Received with EP Mismatch Applies to periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. For OUT endpoints, this bit is reserved.	0x0
[4]	R/W	INTKNTXFEMP	IN Token Received When TxFIFO is Empty Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO was empty. This interrupt is asserted on the endpoint	0x0

Bit	R/W	Symbol	Description	Reset Value
			for which the IN token was received.	
[3]	R/W	TIMEOUT	Timeout Condition Applies to non-isochronous IN endpoints only. Indicates that the core has detected a timeout condition on theUSB for the last IN token on this endpoint.	0x0
[2]	R/W	AHBERR	AHB Error This is generated only in Internal DMA mode if there is an AHBBrror during an AHB read/write. The application reads thecorresponding endpoint DMA address register to get the erroraddress.	0x0
[1]	R/W	EPDISBLD	Endpoint Disabled Interrupt Thisbit indicates that the endpoint is disabled per the application'srequest.	0x0
[0]	R/W	XFERCOMPL	Transfer Completed Interrupt Indicates that the programmed transfer is complete on the AHBas well as on the USB, for this endpoint.	0x0
DIEPTSIZE0 Address : 0xC004_0910 Reset Value : 0x00000000				
[31:21]	-	RESERVED	Reserved	-
[20:19]	R/W	PKTCNT	Packet Count Indicates the total number of USB packets that constitute theTransfer Size amount of data for endpoint 0. This field is decremented every time a packet is read from theTxFIFO.	0x0
[18:7]	-	RESERVED	Reserved	-
[6:0]	R/W	XFERSIZE	Transfer Size Indicates the transfer size in bytes for endpoint 0. The coreinterrupts the application only after it has exhausted the transfersize amount of data. The transfer size can be set to themaximum packet size of the endpoint, to be interrupted at theend of each packet. The core decrements this field every time a packet from theexternal memory is written to theTxFIFO.	0x0
DIEPDMA0 Address : 0xC004_0914 Reset Value : 0x00000000				
[31:0]	R/W	DMAADDR	DMA Address Holds the start address of the external memory for storing orfetching endpoint data. This register is incremented on everyAHB transaction. Note: For control endpoints, this address stores control OUTdata packets as well as SETUP transaction data packets. Ifmultiple SETUP packets are received back-to-back, theSETUP data packet in the memory is overwritten.	0x0
DIEPCTLn (1 ≤ n ≤ 15, DIEPCTL1, DIEPCTL2, ..., DIEPCTL15) Address : 0xC004_0900 + n*0x20 Reset Value : 0x00000000				
[31]	R/W	EPENA	Endpoint Enable Applies to IN and OUT endpoints. For IN endpoint, this bit indicates that data is ready to be transmitted on the endpoint. For OUT endpoints, this bit indicates that the application has allocated the memory to start receivingdata from the USB. The core clears this bit before setting any ofthe following interrupts on this endpoint : SETUP Phase Done (OUT only)	0x0

Bit	R/W	Symbol	Description	Reset Value
			Endpoint Disabled Transfer Complete Transfer Completed Note: For control OUT endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.	
[30]	R/W	EPDIS	Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	0x0
[29]	W	SETD1PID SETODDFR	Set DATA1 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. Set Odd (micro) frame Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame field to odd(micro)frame.	0x0
[28]	W	SETD0PID SETEVENFR	Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. Set Even (micro) frame Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro) frame field to even(micro) frame.	0x0
[27]	W	SETNAK	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core also sets this bit for an endpoint after a SETUP packet is received on that endpoint.	0x0
[26]	W	CNAK	Clear NAK A write to this bit clears the NAK bit for the endpoint.	0x0
[25:22]	R/W	TXFNUM	TxFIFO Number Applies to IN endpoints only. Non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number. 0 : Non-Periodic TxFIFO Others : Specified Periodic TxFIFO number Note: An interrupt IN endpoint could be configured as a nonperiodic endpoint for applications like mass storage.	0x0
[21]	R/W	STALL	STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application clears this bit, never the core. Applies to control endpoints only The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	0x0
[20]	-	RESERVED	Reserved	-

Bit	R/W	Symbol	Description	Reset Value
[19:18]	R	EPTYPE	<p>Endpoint Type This is the transfer type supported by this logical endpoint.</p> <p>0 : Control 1 : Isochronous 2 : Bulk 3 : Interrupt</p>	0x0
[17]	R	NAKSTS	<p>NAK Status Indicates the following:</p> <p>0: The core is transmitting non-NAK handshakes based on the FIFO status 1: The core is transmitting NAK handshakes on this endpoint.</p> <p>If either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO.</p> <p>Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>	0x0
[16]	R	DPID EO_FRCNUM	<p>Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. Applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID.</p> <p>0 : DATA0 1 : DATA1 Even/ Odd (Micro) Frame Applies to isochronous IN and OUT endpoints only.</p> <p>Indicates the (micro) frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/ odd (micro) frame number in which it intends to transmit/ receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register.</p> <p>0 : Even (micro)frame 1 : Odd (micro)frame</p>	-
[15]	R/W	USBACTEP	<p>USB Active Endpoint Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.</p>	0x0
[14:11]	R/W	NEXTEP	<p>Next Endpoint Applies to non-periodic IN endpoints only.</p> <p>Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core accesses this field, even if the Endpoint Enable bit is not set. This field is not valid in Slave mode operation.</p>	0x0
[10:0]	R/W	MPS	<p>Maximum Packet Size The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.</p>	0x0

Bit	R/W	Symbol	Description	Reset Value
DIEPINTn (1 ≤ n ≤ 7, DIEPINT1, DIEPINT2, ..., DIEPINT7) Address : 0xC004_0908 + n*0x20 Reset Value : 0x00000080				
[31:7]	-	EPENA \$\$\$	Reserved	0x1
[6]	R	INEPNAKEFF	<p>IN Endpoint NAK Effective</p> <p>Applies to periodic IN endpoints only.</p> <p>Indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This bit is cleared if the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK.</p> <p>This interrupt indicates that the core has sampled the NAK bit set. This interrupt does not necessarily mean that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.</p>	0x0
[5]	R/W	INTKNEPMIS	<p>IN Token Received with EP Mismatch</p> <p>Applies to periodic IN endpoints only.</p> <p>Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received.</p> <p>For OUT endpoints, this bit is reserved.</p>	0x0
[4]	R/W	INTKNTXFEMP	<p>IN Token Received When TxFIFO is Empty</p> <p>Applies to non-periodic IN endpoints only.</p> <p>Indicates that an IN token was received when the associated TxFIFO was empty. This interrupt is asserted on the endpoint for which the IN token was received.</p>	0x0
[3]	R/W	TIMEOUT	<p>Timeout Condition</p> <p>Applies to non-isochronous IN endpoints only.</p> <p>Indicates that the core has detected a timeout condition on theUSB for the last IN token on this endpoint.</p>	0x0
[2]	R/W	AHBERR	<p>AHB Error</p> <p>Applies to IN and OUT endpoints.</p> <p>This is generated only in Internal DMA mode if there is an AHBERror during an AHB read/write. The application reads thecorresponding endpoint DMA address register to get the erroraddress.</p>	0x0
[1]	R/W	EPDISBLD	<p>Endpoint Disabled Interrupt</p> <p>Applies to IN and OUT endpoints.</p> <p>Thisbit indicates that the endpoint is disabled per the application'srequest.</p>	0x0
[0]	R/W	XFERCOMPL	<p>Transfer Completed Interrupt</p> <p>Applies to IN and OUT endpoints.</p> <p>Indicates that the programmed transfer is complete on the AHBas well as on the USB, for this endpoint.</p>	0x0
DIEPTSIZn (1 ≤ n ≤ 7, DIEPTSIZ1, DIEPTSIZ2, ..., DIEPTSIZ7) Address : 0xC004_0910 Reset Value : 0x00000000				
[31]	-	RESERVED	Reserved	-
[30:29]	R/W	MC	Multi Count	-

Bit	R/W	Symbol	Description	Reset Value
			<p>For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints.</p> <p>1 : 1 packet 2 : 2 packets 3 : 3 packets</p> <p>For non-periodic IN endpoints, this field is valid only in InternalDMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-nControl register</p>	
[28:19]	R/W	PKTCNT	<p>Packet Count</p> <p>Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0.</p> <p>This field is decremented every time a packet is read from the Tx FIFO.</p>	0x0
[18:0]	R/W	XFERSIZE	<p>Transfer Size</p> <p>This field contains the transfer size in bytes for the current endpoint. The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size is set to the maximum packet size of the endpoint, to be interrupted at the end of each packet.</p> <p>The core decrements this field every time a packet from the external memory is written to the Tx FIFO.</p>	0x0
DIEPDMAn (1 ≤ n ≤ 7, DIEPDMA1, DIEPDMA2, ..., DIEPDMA7) Address : 0xC004_0914 + n *0x20 Reset Value : 0x00000000				
[31:0]	R/W	DMAADDR	<p>DMA Address</p> <p>Holds the start address of the external memory for storing or fetching endpoint data. This register is incremented on every AHB transaction.</p> <p>Note: For control endpoints, this address stores control OUT data packets as well as SETUP transaction data packets. If multiple SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten.</p>	0x0
DOEPCTL0 Address : 0xC004_0B00 Reset Value : 0x00008000				
[31]	R/W	EPENA	<p>Endpoint Enable</p> <p>Indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint.</p> <p>SETUP Phase Done</p> <p>Endpoint Disabled</p> <p>Transfer Complete</p> <p>Note: In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory. Transfer Completed</p>	0x0
[30]	R/W	EPDIS	<p>Endpoint Disable</p> <p>The application cannot disable control OUT endpoint 0.</p>	0x0
[29:28]	-	RESERVED	Reserved	-
[27]	W	SETNAK	<p>Set NAK</p> <p>A write to this bit sets the NAK bit for the endpoint.</p> <p>Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core also sets this bit for an endpoint after a SETUP packet is received on that endpoint.</p>	0x0

Bit	R/W	Symbol	Description	Reset Value
[26]	W	CNAK	Clear NAK A write to this bit clears the NAK bit for the endpoint.	0x0
[25:22]	-	RESERVED	Reserved	-
[21]	R/W	STALL	STALL Handshake The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit or Global OUTNAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	0x0
[20]	R/W	SNP	Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	-
[19:18]	R	EPTYPE	Endpoint Type Hardcoded to 00 for control	0x0
[17]	R	NAKSTS	NAK Status Indicates the following: 0 : The core is transmitting non-NAK handshakes based on the FIFO status 1 : The core is transmitting NAK handshakes on this endpoint If this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the Tx FIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	0x0
[16]	-	RESERVED	Reserved	-
[15]	R	USBACTEP	USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.	0x1
[14:2]	-	RESERVED	Reserved	-
[1:0]	R	MPS	Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. 0 : 64 bytes 1 : 32 bytes 2 : 16 bytes 3 : 8 bytes	0x0
DOEPINT0				
Address : 0xC004_0B08				
Reset Value : 0x00000000				
[31:7]	-	EPENA	Reserved	0x1
[6]	R	BACK2BACKSETUP	Back-to-Back SETUP Packets Received This bit indicates that core has received more than three back-to-back SETUP packets for this particular endpoint.	0x0
[5]	-	RESERVED	Reserved	-
[4]	R/W	OUTTKNEPDIS	Token Received When Endpoint Disabled Indicates that an OUT token is received if the endpoint is not enabled. This interrupt is asserted on the endpoint for which the OUT token was received.	0x0

Bit	R/W	Symbol	Description	Reset Value
[3]	R/W	SETUP	SETUP Phase Done Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application decodes the received SETUP data packet.	0x0
[2]	R/W	AHBERR	AHB Error This is generated only in Internal DMA mode if there is an AHBERror during an AHB read/write. The application reads thecorresponding endpoint DMA address register to get the erroraddress.	0x0
[1]	R/W	EPDISBLD	Endpoint Disabled Interrupt Thisbit indicates that the endpoint is disabled per the application'srequest.	0x0
[0]	R/W	XFERCOMPL	Transfer Completed Interrupt Indicates that the programmed transfer is complete on the AHBas well as on the USB, for this endpoint.	0x0
DOEPTSIZE Address : 0xC004_0B10 Reset Value : 0x00000000				
[31]	-	RESERVED	Reserved	-
[30:29]	R/W	SUPCNT	SETUP Packet Count This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 1 : 1 packet 2 : 2 packets 3 : 3 packets	-
[28:20]	-	RESERVED	Reserved	-
[19]	R/W	PKTCNT	Packet Count This field is decremented to zero after a packet is written intothe RxFIFO.	0x0
[18:7]	-	RESERVED	Reserved	-
[6:0]	R/W	XFERSIZE	Transfer Size Indicates the transfer size in bytes for endpoint 0. The coreinterrupts the application only after it has exhausted thetransfer size amount of data. The transfer size can be set tothe maximum packet size of the endpoint, to be interrupted atthe end of each packet. The core decrements this field every time a packet is read fromRxFIFO and written to the external memory.	0x0
DOEPDMA0 Address : 0xC004_0B14 Reset Value : 0x00000000				
[31:0]	R/W	DMAADDR	DMA Address Holds the start address of the external memory for storing orfetching endpoint data. This register is incremented on everyAHB transaction. Note: For control endpoints, this address stores control OUTdata packets as well as SETUP transaction data packets. Ifmultiple SETUP packets are received back-to-back, theSETUP data packet in the memory is overwritten.	0x0
DOEPCTLn (1 ≤ n ≤ 15, DOEPCtl1, DOEPCtl2, ..., DOEPCtl15) Address : 0xC004_0B00 + n*0x20 Reset Value : 0x00000000				

Bit	R/W	Symbol	Description	Reset Value
[31]	R/W	EPENA	<p>Endpoint Enable For IN endpoint, this bit indicates that data is ready to be transmitted on the endpoint. For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint:</p> <p>SETUP Phase Done (OUT only) Endpoint Disabled Transfer Complete Transfer Completed Note: For control OUT endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.</p>	0x0
[30]	R/W	EPDIS	<p>Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete.</p> <p>The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.</p>	0x0
[29]	W	SETD1PID SETODDFR	<p>Set DATA1 PID Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.</p> <p>Set Odd (micro) frame Applies to isochronous IN and OUT endpoints only.</p> <p>Writing to this field sets the Even/Odd (micro) frame field to odd(micro)frame.</p>	0x0
[28]	W	SETD0PID SETEVENFR	<p>Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only.</p> <p>Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.</p> <p>Set Even (micro) frame Applies to isochronous IN and OUT endpoints only.</p> <p>Writing to this field sets the Even/Odd (micro) frame field to even(micro) frame.</p>	0x0
[27]	W	SETNAK	<p>Set NAK A write to this bit sets the NAK bit for the endpoint.</p> <p>Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core also sets this bit for an endpoint after a SETUP packet is received on that endpoint.</p>	0x0
[26]	W	CNAK	<p>Clear NAK A write to this bit clears the NAK bit for the endpoint.</p>	0x0
[25:22]	-	RESERVED	Reserved	-
[21]	R/W	STALL	<p>STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only.</p> <p>The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application clears this bit, never the core.</p> <p>Applies to control endpoints only</p> <p>The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>	0x0
[20]	R/W	SNP	<p>Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode,</p>	-

Bit	R/W	Symbol	Description	Reset Value
			the core does not check the correctness of OUT packets before transferring them to application memory.	
[19:18]	R	EPTYPE	<p>Endpoint Type This is the transfer type supported by this logical endpoint.</p> <p>0 : Control 1 : Isochronous 2 : Bulk 3 : Interrupt</p>	0x0
[17]	R	NAKSTS	<p>NAK Status Indicates the following:</p> <p>0: The core is transmitting non-NAK handshakes based on the FIFO status 1: The core is transmitting NAK handshakes on this endpoint. If either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.</p>	0x0
[16]	R	DPID EO_FRNUM	<p>Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. Applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID.</p> <p>0 : DATA0 1 : DATA1 Even/Odd (Micro) Frame Applies to isochronous IN and OUT endpoints only. Indicates the (micro) frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register.</p> <p>0 : Even (micro)frame 1 : Odd (micro)frame</p>	-
[15]	R/w	USBACTEP	USB Active Endpoint Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.	0x0
[14:11]	-	RESERVED	Reserved	-
[10:0]	R/W	MPS	<p>Maximum Packet Size The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.</p>	0x0
DOEPINTn (1 ≤ n ≤ 7, DOEPINT1, DOEPINT2, ..., DOEPINT7)				
Address : 0xC004_0B08 + n*0x20				
Reset Value : 0x00000000				

Bit	R/W	Symbol	Description	Reset Value
[31:7]	-	\$\$\$ EPENA	Reserved	0x1
[6]	R	BACK2BACKSETUP	Back-to-Back SETUP Packets Received This bit indicates that core has received more than three back-to-back SETUP packets for this particular endpoint.	0x0
[5]	-	RESERVED	Reserved	-
[4]	R/W	OUTTKNEPDIS	Token Received When Endpoint Disabled Indicates that an OUT token is received if the endpoint is not enabled. This interrupt is asserted on the endpoint for which the OUT token was received.	0x0
[3]	R/W	SETUP	SETUP Phase Done Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application decodes the received SETUP data packet.	0x0
[2]	R/W	AHBERR	AHB Error This is generated only in Internal DMA mode if there is an AHB error during an AHB read/write. The application reads the corresponding endpoint DMA address register to get the error address.	0x0
[1]	R/W	EPDISBLD	Endpoint Disabled Interrupt This bit indicates that the endpoint is disabled per the application's request.	0x0
[0]	R/W	XFERCOMPL	Transfer Completed Interrupt Indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.	0x0
DOEPTSIZn (1 ≤ n ≤ 7, DOEPTSIZ1, DOEPTSIZ2, ..., DOEPTSIZ7)				
Address : 0xC004_0B10				
Reset Value : 0x00000000				
[31]	-	RESERVED	Reserved	-
[30:29]	R/W	RXDPID SUPCNT	Received Data PID This is the data PID received in the last packet for this endpoint. 0: DATA0 1: DATA1 2: DATA2 3: MDATA SETUP Packet Count This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 1: 1 packet 2: 2 packets 3: 3 packets	-
[28:19]	R/W	PKTCNT	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. This field is decremented every time a packet is read from the Tx FIFO.	0x0
[18:0]	R/W	XFERSIZE	Transfer Size This field contains the transfer size in bytes for the current endpoint. The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size is set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet from the external memory is written to	0x0

Bit	R/W	Symbol	Description	Reset Value
			the TxFIFO.	
DOEPDMA_n (1 ≤ n ≤ 7, DOEPDMA1, DOEPDMA2, ..., DOEPDMA7)				
<i>Address : 0xC004_0B14 + n*0x20</i>				
Reset Value : 0x00000000				
[31:0]	R/W	DMAADDR	<p>DMA Address</p> <p>Holds the start address of the external memory for storing or fetching endpoint data. This register is incremented on every AHB transaction.</p> <p>Note: For control endpoints, this address stores control OUTdata packets as well as SETUP transaction data packets. If multiple SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten.</p>	0x0

25.4.6 Power and Clock Gating Registers (USB_OTG_PCGCCTL)

Bit	R/W	Symbol	Description	Reset Value
PCGCCTL				
<i>Address : 0xC004_0E00</i>				
Reset Value : 0x00000000				
[31:1]	-	RESERVED	Reserved	-
[0]	R/W	STOPPCLK	<p>STOP Pclk</p> <p>The application sets this bit to stop the PHY clock if the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit if the USB is resumed or a new session starts.</p>	

Section 26. **USB2.0 HOST**

26.1 Overview

The USB 2.0 EHCI Host Controller is fully compliant with the USB 2.0 specification and the Enhanced Host Controller Interface (EHCI) specification (revision 2.0). The controller supports high-speed, 480Mbps transfers (40 times faster than USB 1.1 full-speed mode) as well as companion controller integration with the USB 1.1 OHCI Host Controller. The controller is designed to operate independently of the Bus Interface Unit (BIU) to the Application, shielding the complexities of the USB 2.0 Host Controller native protocol and providing easy integration of the EHCI Host Controller with an industry-standard AHB or PCI bus or with your target application. At the USB 2.0 physical interface, the EHCI Host Controller is designed with USB 2.0 Transceiver Macrocell Interface. Also the controller provides High-Speed Inter-Chip(HSIC)(version1.0).

26.1.1 Features

The host controller is responsible for:

- Detecting the attachment and removal of USB devices
- Collecting status and activity statistics
- Controlling power supply to attached USB devices
- Controlling the association to either the Open Host Controller Interface or the Enhanced Host Controller via a Port Router
- Root Hub functionality to support up/down stream port
- Support High-Speed Inter-Chip (HSIC), Version 1.0

26.1.2 Block Diagram

The architecture of the USB 2.0 EHCI Host Controller with BIU (Bus Interface Unit), along with the major building blocks and the companion controller (USB 1.1 OHCI Host Controller), is shown in Figure 26-1

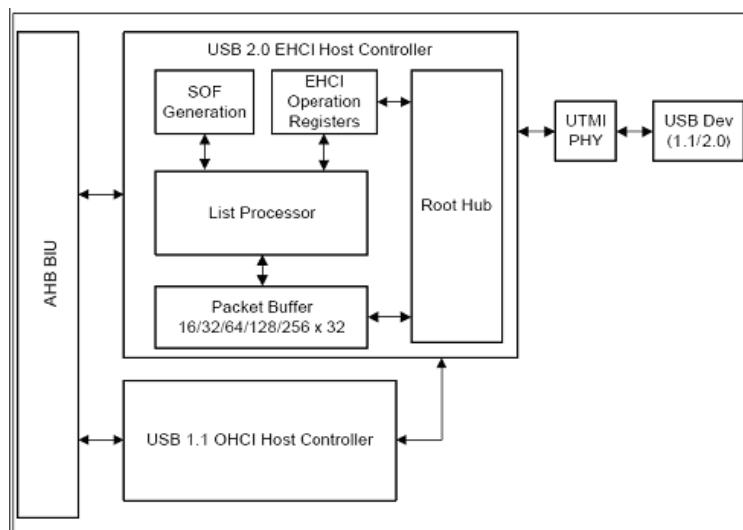


Figure 26-1. USB2.0 HOST Block Diagram

26.2 Functional Description

26.2.1 Programming User Config of PHY and LINK in EHCI or OHCI

- 1) Release common reset of host controller
 - Program RSTCON1[24](addr: 0xC0012004) to 1'b1
- 2) Progarm AHB Burst type
 - SINGLE : default. TIEOFFREG7[27:25](addr: 0xC001101C) is 3'b000.
 - INCR16(The OHCI does not support INCR16) : program TIEOFFREG7[27:25](addr: 0xC001101C) to 3'b111(recommended)
 - INCR8(The OHCI does not support INCR8) : program TIEOFFREG7[27:25](addr: 0xC001101C) to 3'b110
 - INCR4(The OHCI part of the controller only supports INCR4 or SINGLE) : program TIEOFFREG7[27:25](addr: 0xC001101C) to 3'b100
- 3) Select word interface and enable word interface selection
 - 8bit word interface : program TIEOFFREG5[26:25](addr: 0xC0011014) to 2'b01 and TIEOFFREG9[10:9](addr: 0xC0011024) to 2'b01
 - 16bit word interface : program TIEOFFREG5[26:25](addr: 0xC0011014) to 2'b11 and TIEOFFREG9[10:9](addr: 0xC0011024) to 2'b11
- 4) POR(Power On Reset) of PHY
 - Program TIEOFFREG8[8:7](addr: 0xC0011020) to 2'b01
- 5) Wait clock of PHY – about 40 micro seconds
- 6) Release utmi reset
 - Program TIEOFFREG5[21:20](addr: 0xC0011014) to 2'b11
- 7) Release ahb reset of EHCI, OHCI
 - Program TIEOFFREG13[19:17](addr: 0xC0011034) to 3'b111

26.2.2 Programming User Config of PHY and LINK in HSIC

- 1) Release common reset of host controller
 - Program RSTCON1[24](addr: 0xC0012004) to 1'b1
- 2) Progarm AHB Burst type
 - SINGLE : default. TIEOFFREG7[27:25](addr: 0xC001101C) is 3'b000.
 - INCR16(The OHCI does not support INCR16) : program TIEOFFREG7[27:25](addr: 0xC001101C) to 3'b111
 - INCR8(The OHCI does not support INCR8) : program TIEOFFREG7[27:25](addr: 0xC001101C) to 3'b110
 - INCR4(The OHCI part of the controller only supports INCR4 or SINGLE) : program TIEOFFREG7[27:25](addr: 0xC001101C) to 3'b100
- 3) Select word interface and enable word interface selection

- 8bit word interface : program TIEOFFREG5[26:25](addr: 0xC0011014) to 2'b01 and TIEOFFREG9[10:9](addr: 0xC0011024) to 2'b01 and TIEOFFREG11[13:12](addr: 0xC001102C) to 2'b01
 - 16bit word interface : program TIEOFFREG5[26:25](addr: 0xC0011014) to 2'b11 and TIEOFFREG9[10:9](addr: 0xC0011024) to 2'b11 and TIEOFFREG11[13:12](addr: 0xC001102C) to 2'b11
- 4) POR(Power On Reset) of EHCI PHY
- Program TIEOFFREG8[8:7](addr: 0xC0011020) to 2'b01
- 5) Wait clock of EHCI PHY – about 40 micro seconds
- 6) Program HSIC Mode
- Program TIEOFFREG5[24:23](addr: 0xC0011014) to 2'b11
- 7) POR(Power On Reset) of HSIC PHY
- Program TIEOFFREG10[19:18](addr: 0xC0011028) to 2'b01
- 8) Wait clock of HSIC PHY – about 40 micro seconds
- 9) Release utmi reset
- Program TIEOFFREG5[20](addr: 0xC0011014) to 1'b1 and TIEOFFREG5[22](addr: 0xC0011014) to 1'b1
- 10) Release ahb reset
- Program TIEOFFREG13[19:17](addr: 0xC0011034) to 3'b111

26.2.3 Attention point of HSIC programming

- Port Status Control Register:
 - EHCI uses PORTSC_1 (addr: 0xC0030054)
 - HSIC uses PORTSC_2 (addr: 0xC0030058)
- How to enable the Port Power Control Switch:
 - EHCI case: set PORTSC_1[12](addr: 0xC0030054) to 1'b1
 - HSIC case: set INSNREG08[1](addr: 0x0xC00300B0) to 1'b1

26.3 Register Summary

■ Base Address

- OHCI controller: 0xC0020000
- EHCI controller: 0xC0003000.

The USB2.0 Host controller implements all of the necessary EHCI registers. Consult the Enhanced Host Controller Interface Specification for USB for register details and programming considerations. Registers that differ from the EHCI specification are described below.

Bit	R/W	Symbol	Description	Reset Value
HCCAPBASE (HCCAPBASE)				
<i>Address : C003_0000h</i>				
[31:0]	R/W	HCCAPBASE	Capability Register	32'h0100_00 10
HCSPARAMS (HCSPARAMS)				
<i>Address : C003_0004h</i>				
[31:0]	R/W	HCSPARAMS	Structural Parameter Register	32'h0000_11 16
HCCPARAMS (HCCPARAMS)				
<i>Address : C003_0008h</i>				
[31:0]	R/W	HCCPARAMS	Capability Parameter Register Note : The Isochronous Scheduling Threshold value is set to 1 by default	32'h0000_A 010
RESERVED				
<i>Address : C003_000Ch</i>				
USBCMD (USBCMD)				
<i>Address : C003_0010h</i>				
[31:0]	R/W	USBCMD	USB Command	32'h0008_00 00
USBSTS (USBSTS)				
<i>Address : C003_0014h</i>				
[31:0]	R/W	USBSTS	USB Status	32'h0000_10 00
USBINTR (USBINTR)				
<i>Address : C003_0018h</i>				
[31:0]	R/W	USBINTR	USB Interrupt Enable	32'h0000_00 00
FRINDEX (FRINDEX)				
<i>Address : C003_001Ch</i>				
[31:0]	R/W	FRINDEX	USB Frame Index	32'h0000_00 00
CTRLDSSEGMEN (CTRLDSSEGMEN)				
<i>Address : C003_0020h</i>				
[31:0]	R/W	CTRLDSSEGMEN	4G Segment Selector	32'h0000_00

Bit	R/W	Symbol	Description	Reset Value	
				00	
PERIODICLISTBASE (PERIODICLISTBASE)					
<i>Address : C003_0024h</i>					
[31:0]	R/W	PERIODICLISTBASE	Periodic Frame List Base Address Register	32'h0000_0000	
ASYNCLISTADDR (ASYNCLISTADDR)					
<i>Address : C003_0028h</i>					
[31:0]	R/W	ASYNCLISTADDR	Asynchronous List Address Register	32'h0000_0000	
RESERVED					
<i>Address : C003_002Ch ~ C003_004Ch</i>					
CONFIGFLAG (CONFIGFLAG)					
<i>Address : C003_0050h</i>					
[31:0]	R/W	CONFIGFLAG	Configured Flag Register	32'h0000_0000	
PORTSC_1to15 (PORTSC_1to15)					
<i>Address : C003_0054h ~ C003_008Ch</i>					
[31:0]	R/W	PORTSC_1TO15	Port Status / Control	32'h0000_2000	
INSNREG00 (INSNREG00)					
<i>Address : Base Address + 0090H : WORD</i>					
[31:14]	-	RESERVED	Reserved	32'h0000_0000	
[13:1]	R/W	BIT8	This value is used as the 1-microframe counter with byte interface (8-bits)		
[12:1]	R/W	BIT16	This value is used as the 1-microframe counter with byte interface (8-bits)		
[0]	R/W	WRENB	Write Enable Register 1 : Enable 0 : Disable		
INSNREG01 (INSNREG01)					
<i>Address : Base Address + 0094H : WORD</i>					
[31:16]	R/W	OUTTHRESHOLD	Programmable Packet Buffer Out Thresholds	32'h0020_0000	
[15:0]	R/W	INTTHRESHOLD	Programmable Packet Buffer IN Threshold	20	
INSNREG02 (INSNREG02)					
<i>Address : Base Address + 0098H : WORD</i>					
[31:0]	R/W	INSNREG02	Programmable Packet Buffer Depth The value specified here is the number of DWORDs (32bit entries)	32'h0000_0080	
INSNREG03 (INSNREG03)					
<i>Address : Base Address + 009CH : WORD</i>					
[31:0]	R/W	INSNREG03	Break Memory Transfer Used in conjunction with INSNREG01 to enable/disable breaking memory transactions into chunks once the OUT/IN threshold value is reached. 1 : Enable 0 : Disable	32'h0000_0000	
INSNREG04 (INSNREG04)					
<i>Address : Base Address + 00A0H : WORD</i>					
[31:5]	-	RESERVED	Reserved	32'h0000_0000	
[4]	R/W	NAK	This value is used as the 1-microframe counter with byte interface (8-bits)		

Bit	R/W	Symbol	Description	Reset Value
[3]	-	RESERVED	Reserved	
[2]	R/W	SCALESDOWN	Scales down port enumeration time enable 1 : Enable 0 : Disable	
[1]	R/W	M_HCCPARAMS	Makes the HCCPARAMS register Write Enable 1: Enable 0: Disable	
[0]	R/W	M_HCSPARAMS	Makes the HCSPARAMS register Write Enable 1: Enable 0: Disable	

INSNREG05 (INSNREG05)

Address : Base Address + 00A4H : WORD

UTMI Configuration*Control and Status Register. Allows the user to read the UTMI registers from the following*

[31:18]	-	RESERVED	Reserved	32'h0000_1000
[17]	R/W	VBUSY	VBusy (Software RO). Hardware indicator that a write to this register has occurred and the hardware is currently processing the operation defined by the data written. When processing is finished, this bit is cleared.	
[16:13]	R/W	VPORT	VPort (Software R/W)	
[12]	R/W	VCONTROL_LOADM	VControlLoadM 1: NOP, (Software R/W) 0 : Load	
[11:8]	R/W	VCONTROL	VControl (Software R/W)	
[7:0]	R/W	VSTATUS	VStatus (Software R/W)	

INSNREG06 (INSNREG06)

Address : Base Address + 00A8H : WORD

AHB Error Status**INSNREG07 (INSNREG07)**

Address : Base Address + 00ACH : WORD

AHB Master Error Address**INSNREG08 (INSNREG08)**

Address : Base Address + 00A4H : WORD

HSIC Enable/Disable

[31:16]	R/W	RESERVED	Reserved for future use. You don't have to write any value except 0.	16'b0
[15:0]	R/W	HSIC_ENB	This register has R/W access to the host driver and gives control to the host driver to enable/disable the HSIC interface per port. Each bit in this register controls the HSIC interface for a particular port. Bit 1 controls PORT 1, Bit 0 controls the HSIC interface for PORT 0, and so on. 0: When HSIC support is selected, then a value of 0 on this control register bit will put the corresponding PORT in the HSIC Disabled state 1: When HSIC configuration is selected and a value of 1 in this control bit will put the corresponding port in HSIC Enabled state	16'b0

Section 27. I2S

27.1 Overview

The I2S controller supports streaming serial audio data between the External I2S Codec (ADC/DAC) and the processor. It consists of one transmitter channel and one receive channel. Both the transmit and receive channel are independent of each other and can work simultaneously.

- The I2S controller interface can be configured to work in three modes by the IMS (Internal Master/Slave) Fields in the I2SMOD register.
 - Internal Master mode
In the Internal master mode the Clock source to the I2S controller is APB 'PCLK'. The Bit Clock 'I2SBCLK' and the word select signal 'I2SLRCLK' is generated internally from PCLK.
 - External Master mode
In the External master mode, the Clock source to the I2S controller is External Codec Clock. The Bit Clock 'I2SBCLK' and the word select signal 'I2SLRCLK' is generated internally from PCLK.
 - Slave mode
In the Slave mode, the Clock source to the I2S controller is APB 'PCLK'. The Bit Clock 'I2SBCLK' and the word select signal 'I2SLRCLK' is also supplied to the I2S Controller From an External source.
- BLC bits in the I2SMOD register can be configured for 8 bit or 16 bit or 24 bit per channel.
- LRP bit in the I2S mod register can be configured to configure the left right word clock polarity.
- SDF in the I2SMOD register can be configured to change the data bit position
 - Internal Master mode
 - MSB (Left) Justified
 - LSB (Right) Justified
- TXR in the I2SMOD register can be configured for half duplex or full duplex transmission.
 - Transmit only mode
 - Receive only mode
 - Transmit receive simultaneous mode
 - No operation / reserved
- DMA transfer mode can be set by setting the TXDMAACTIVE and RXDMAACTIVE bits in the I2SCON register.

27.1.1 Features

- Single Transmit and Receive Channel.
- Supports 8/16/24 bit word length.
- Programmable left/right word clock polarity
- Programmable MSB justified (Left), I2S and LSB (Right justified) data bit position.
- Programmable Transmit/Receive mode and simultaneous transmit receive mode.
- DMA transfer mode.

27.2 Functional Description

The I2S Controller is very feature rich and works in slave only and External master and internal master mode as Transmitter or Receiver. In the Master mode, the Output Codec Clock to the External codec and the BCLKm is generated by the I2SLKCON block. The Audio Channel Clock is then derived from the BCLKm in the I2S Channel generator Block.

In the Slave Mode BITCLK and LRCLK is supplied from an external master.

The I2S controller after the power on reset needs to be configured for I2S Access. The Configuration block is I2SREG block. Here using the APB access the TX and Receive FIFO needs to be flushed before any valid I2S Transaction can occur. Then I2S Data is written using the PWRITE and PWDATA APB signals, which are written onto the TXFIFO block .For the I2S to transmit, the I2SACTIVE Signal should be activated for the TXFIFO block to transfer the data onto I2S Channel Generator Block.

The I2S Channel Generator Block then takes the BITCLK and generates the Channel Clock and the Parallel to serial load transfer control signals to the TX SFTR Block. The Holding Register inside the TX SFTR holds the incoming Parallel data and passes it onto I2SSDO bit by bit aligned with BITCLK. The Channel Clock LRCLK is also passed along with Channel Clock enable.

During the Receive phase, the change on Channel clock is interpreted as the incoming data on the I2SSDI line and is latched in on the positive edge of BCLK inside the RX SFTR.

The I2SACTIVE with the help of BCLK and LRCLK help generate the control signal, Serial to Parallel load in the Channel generator block. The Channel Generator block waits for the Left Channel data and the Right Channel data and passes it onto the RX FIFO Block where it is stored in subsequent locations .Any read from the APB is then provided the stored data.

The DMA Access to the I2S Controller can be activated by enabling the TXDMAACTIVE /RXDMAACTIVE bit in the I2SREG block. The TXDREQ is generated in the TXDMAFSM Block when TXDMAACTIVE is high and the TXFIFO is not full .The RXDREQ is generated in the RXDMAFSM when the RXDMAACTIVE is high and the RX FIFO is not empty. Both requests work on a Request –ACK mode.

A high-level block diagram of the I2S Controller is shown in Figure 27-1.

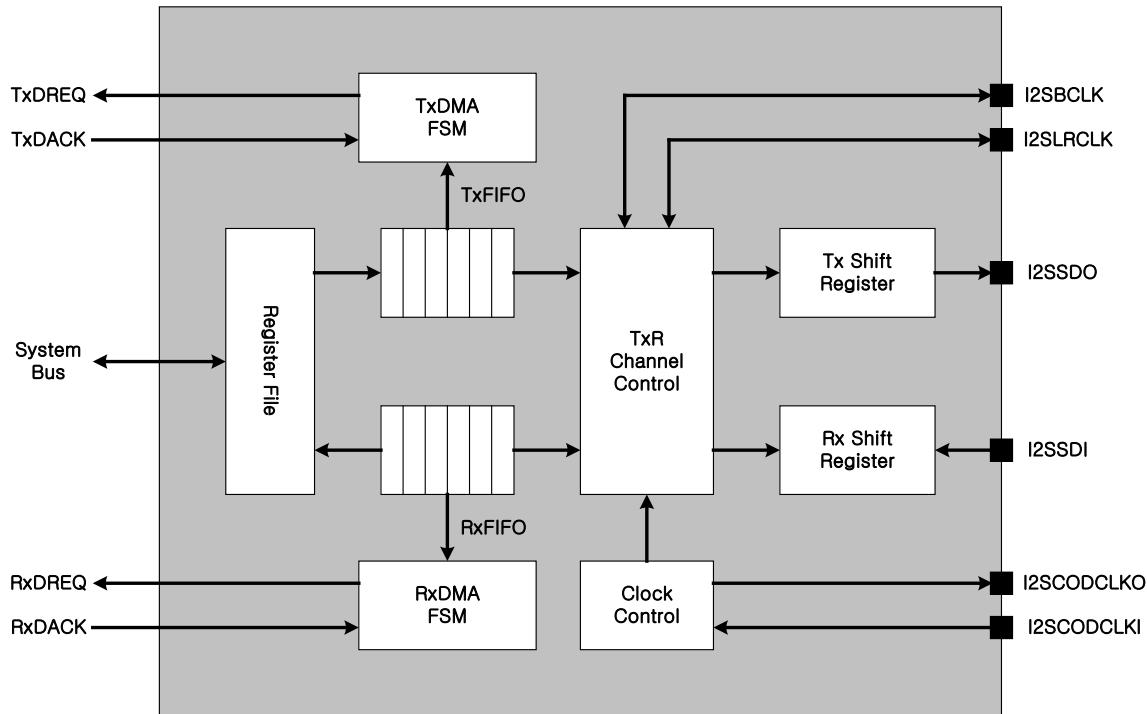


Figure 27-1 Block Diagram of the I2S Controller

27.2.1 Basic Clock Tree

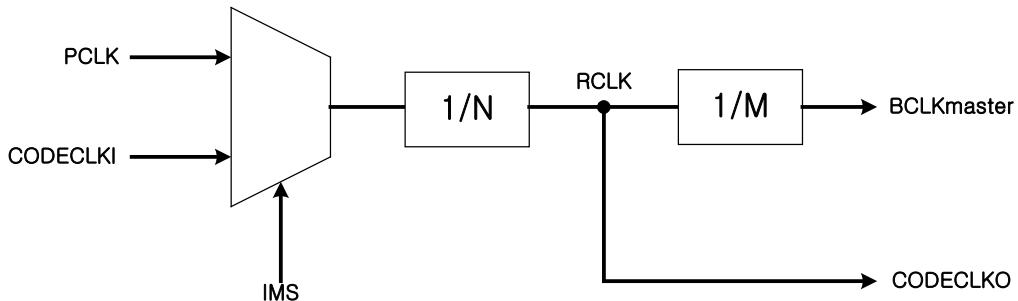


Figure 27-2 Basic Clock Tree

NOTE 1: PSVALA is the value written to the prescalar division register.

NOTE2: PSVALC (internal register in I2SCLKCON) depends upon the RFS and BFS value combined to decide the division value.

For more information please refer to the I2SMOD register and I2S clock mapping table.

27.2.2 I2S-bus format

The I2S bus has four lines including serial data input I2SSDI, serial data output I2SSDO, left/right channel select clock I2SLRCLK, and serial bit clock I2SBCLK; the device generating I2SLRCLK and I2SBCLK is the master.

Serial data is transmitted in 2's complement with the MSB first with a fixed position, whereas the position of the LSB depends on the word length. The transmitter sends the MSB of the next word at one clock period after the I2SLRCLK is changed. Serial data sent by the transmitter may be synchronized with either the trailing or the leading edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so

there are some restrictions when transmitting data that is synchronized with the leading edge.

The LR channel select line indicates the channel being transmitted. I2SLRCLK may be changed on either a trailing or leading edge of the serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The I2SLRCLK line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next word.

- MSB (Left) Justified

MSB-Justified (Left-Justified) format is similar to I2S bus format, except the in MSB-justified format, the transmitter always sends the MSB of the next word at the same time whenever the I2SLRCLK is changed.

- LSB (Right) Justified

LSB-Justified (Right-Justified) format is opposite to the MSB-justified format. In other word, the transferring serial data is aligned with ending point of I2SLRCLK transition.

Formats show the audio serial format of I2S, MSB-justified, and LSB-justified. Note that in this figure, the word length is 16 bit and I2SLRCLK makes transition every 24 cycle of I2SBCLK (BFS is 48 fs, where fs is sampling frequency; I2SLRCLK frequency).

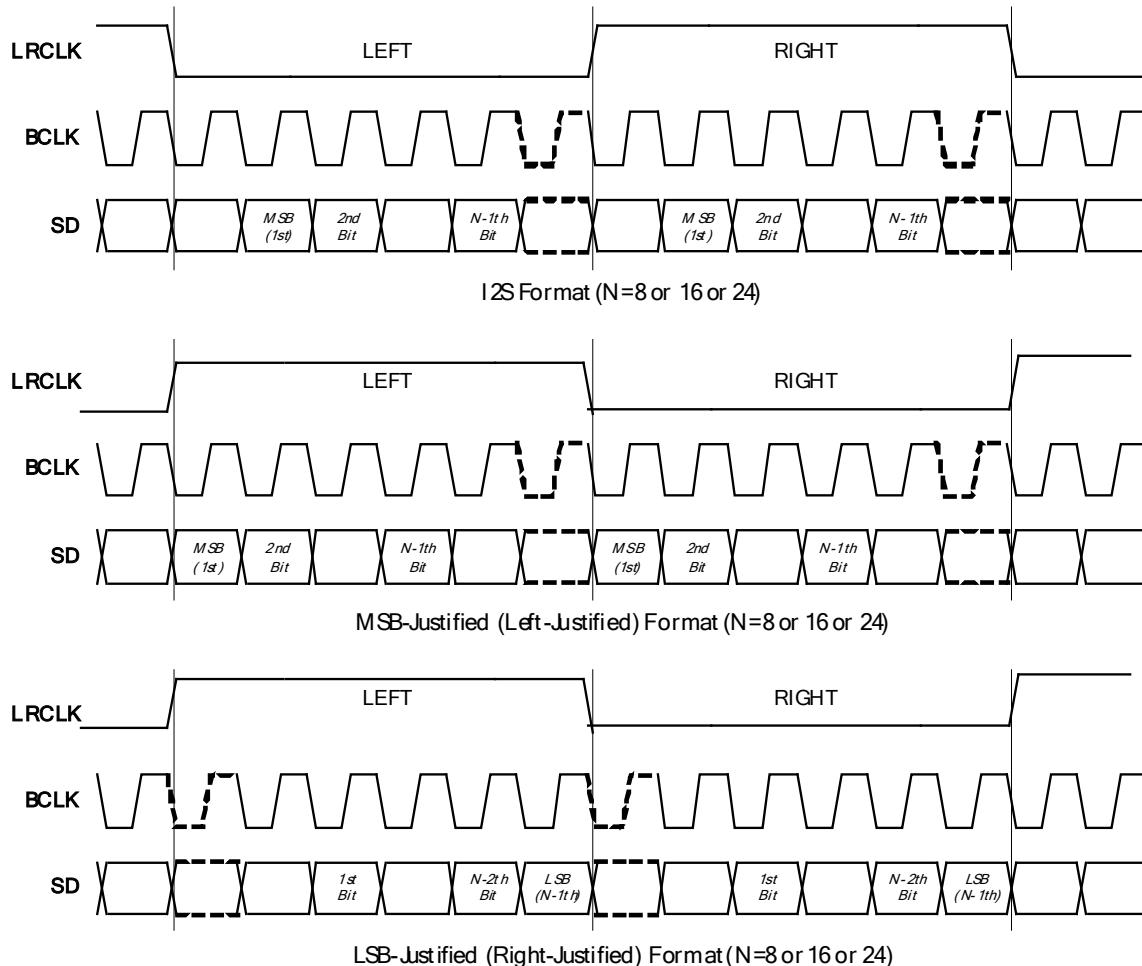


Figure 27-3 I2S Audio Serial Data Formats

27.2.3 Sampling frequency and master clock

Master clock frequency (PCLK) can be selected by sampling frequency as shown in Table 27-1. Because PCLK is made by I2S prescaler, the prescaler value and PCLK type (256 or 384fs) should be determined properly.

I2SLRCLK (fs)	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
CODECLK (MHz)	256fs									
	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
	384fs									
3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640	

Table 27-1 I2S CODEC clock (CODECLK = 256 or 384 fs)

27.2.4 I2S clock mapping table

On selecting BFS, RFS, and BLC bits of I2SMOD register, user should refer to the following table. Table 27-2 shows the allowable clock frequency mapping relations.

Clock Frequency		RFS			
		256 fs (00B)	512 fs (01B)	384 fs (10B)	768 fs (11B)
BFS	16 fs (10B)	(a)	(a)	(a)	(a)
	24 fs (11B)	-	-	(a)	(a)
	32 fs (00B)	(a) (b)	(a) (b)	(a) (b)	(a) (b)
	48 fs (01B)	-	-	(a) (b) (c)	(a) (b) (c)
Description		(a) Allowed when BLC is 8-bit (b) Allowed when BLC is 16-bit (c) Allowed when BLC is 24-bit			

Table 27-2 I2S clock mapping table

27.3 Programming Guide

27.3.1 Tx Channel

The I2S TX channel provides a single stereo compliant output. The transmit channel can operate in master or Slave mode. Data is transferred between the processor and the I2S controller via an APB access or a DMA access.

The processor must write words in multiples of two (i.e. for left and right audio sample). The words are serially shifted out timed with respect to the audio bitclk, BCLK and word select clock, LRCLK.

TX Channel has 16X32 bit wide FIFO where the processor or DMA can write upto 16 left/right data samples after enabling the channel for transmission.

An Example sequence is as the following.

Ensure the PCLK and CODCLKI are coming correctly to the I2S controller and FLUSH the TX FIFO using the TFLUSH bit in the I2SFIC Register (I2S FIFO Control Register).

Please ensure that I2S Controller is configured in one of the following modes

- TX only mode
- TX/RX simultaneous mode.

This can be done by programming the TXR bit in the I2SMOD register (I2S mode register).

Then program the following parameters according to the need

- IMS
- SDF
- BFS
- BLC
- LRP

For programming, the above-mentioned fields please refer I2SMOD register (I2S mode register).

Once ensured that the input clocks for I2S controller are up and running and step 1 and 2 have been completed we can write to TX FIFO.

The write to the TX FIFO has to be carried out through the I2STXD Register (I2S TX FIFO Register). This 32 bit data will occupy position 0 of the FIFO and any further data will be written to position 2, 3 and so on.

The Data is aligned in the TX FIFO for 8-bits/channel or 16-bits/channel BLC as shown

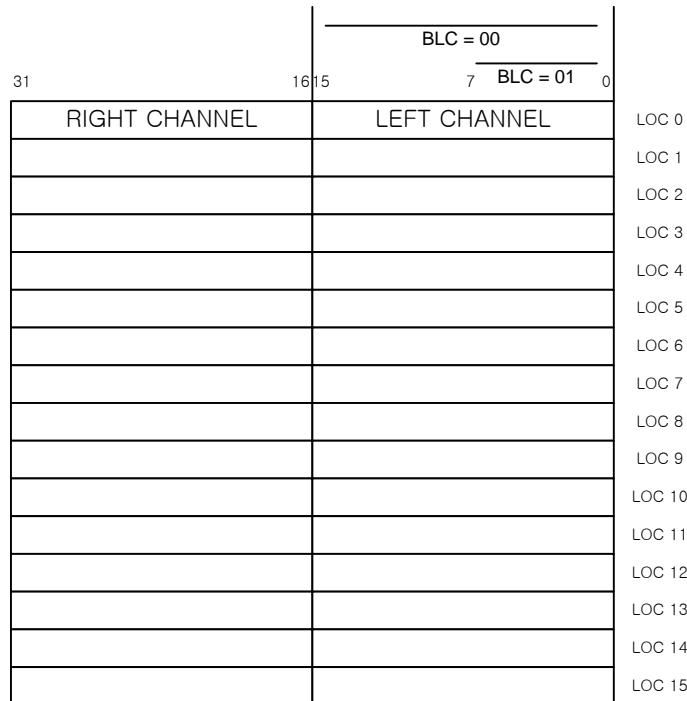


Figure 27-4 TX FIFO Structure for BLC = 00 or BLC = 01

The data is aligned in the TX FIFO for 24-bits / channel BLC as shown.



Figure 27-5 TX FIFO Structure for BLC = 10 (24-bits/channel)

Once the data is written to the TX FIFO the TX channel can be made active by enabling the I2SACTIVE bit in the I2SCON Register (I2S Control Register).The data is then serially shifted out with respect to the bit clock BCLK and word select clock LRCLK.

The TXCHPAUSE in the I2SCON Register (I2S Control Register) can stop the serial data transmission on the I2SSDO.The transmission is stopped once the current Left/Right channel is transmitted.If the control registers in the

I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) are to be reprogrammed then it is advisable to disable the TX channel.

If the TX channel is enabled while the FIFO is empty, no samples are read from the FIFO. The Status of TX FIFO can be checked by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

27.3.2 Rx Channel

The I2S RX channel provides a single stereo compliant output. The receive channel can operate in master or slave mode. Data is received from the input line and transferred into the RX FIFO. The processor can then read this data via an APB read or a DMA access can access this data.

RX Channel has a 16X32 bit wide RX FIFO where the processor or DMA can read up to 16 left/right data samples after enabling the channel for reception.

An Example sequence is as following.

Ensure the PCLK and CODCLKI are coming correctly to the I2S controller and FLUSH the RX FIFO using the RFLUSH bit in the I2SFIC Register (I2S FIFO Control Register) and the I2S controller is configured in any of the modes.

- Receive only.
- Receive / Transmit simultaneous mode.

This can be done by programming the TXR bit in the I2SMOD register (I2S mode register).

Then program the following parameters according to the need.

- IMS
- SDF
- BFS
- BLC
- LRP

For programming, the above mentioned fields please refer I2SMOD register. (I2S mode register).

Once ensured that the input clocks for I2S controller are up and running and step1 and 2 have been completed user must put the I2SACTIVE high to enable any reception of data, the I2S controller receives data on the LRCLK change.

The Data must be read from the RX FIFO using the I2SRXD Register (I2S RX FIFO Register) only after looking at the RX FIFO count in the I2SFIC Register (I2S FIFO Control Register). The count would only increment once the complete left channel and right have been received.

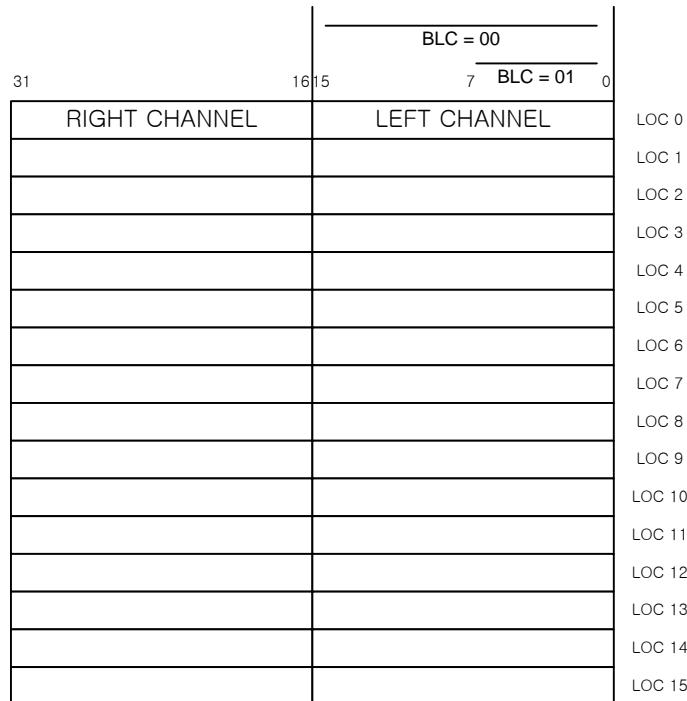


Figure 27-6 RX FIFO Structure for BLC = 00 or BLC = 01

The data is aligned in the RX FIFO for 24-bits/channel BLC as shown.

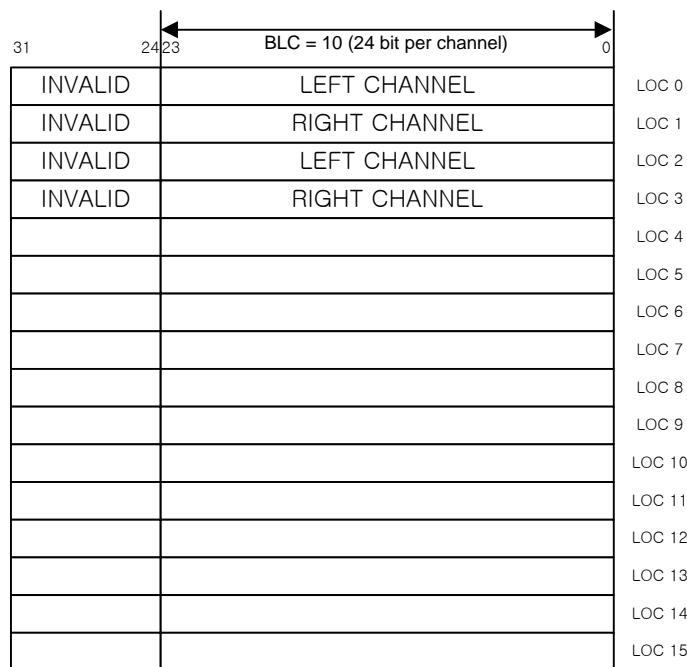


Figure 27-7 RX FIFO Structure for BLC = 10 (24-bits/channel)

The RXCHPAUSE in the I2SCON register can stop the serial data reception on the I2SSDI. The reception is stopped once the current Left/Right channel is received. If the control registers in the I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) are to be reprogrammed then it is advisable to disable the RX channel.

The Status of RX FIFO can be checked by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

27.3.3 I2S-Controller Initialize

The initialization sequence in this section is used for I2S-bus controller.

- Provide a clock and Release a reset. This signals are propagated all of the I2S-bus controller internal registers and logic.
- Enable pad select alt-function of I2S

Following diagram shows I2S-controller operation command examples.

Transmitter Mode

```
// to function in the TX mode

void activate_I2S_controller ( void )
{
    apbif_single_write (SEND_TO_I2S, FIFO_CONTROL_REG, 0xFFFF_FFFF); // Flush the current FIFO
    apbif_single_write (SEND_TO_I2S, FIFO_CONTROL_REG, 0x0000_0000); // Clear the Flush bit
    apbif_multiple_write(SEND_TO_I2S, TX_REG, data_value ); // Data to be transmitted
    apbif_single_write (SEND_TO_I2S, MODE_REG, 0x0000_0000 ); //Configure the mode register
    apbif_single_write (SEND_TO_I2S, CONTROL_REG,0x0000_0001 ); // Initiate the transfer
    wait_until_data_is_transmitted( ); // wait for data to be transmitted
    apbif_single_write (SEND_TO_I2S, CONTROL_REG,0x0000_0000 ); // End the transfer
};
```

Receiver Mode

```
void activate_I2S_controller ( void )
{
    apbif_single_write (SEND_TO_I2S, FIFO_CONTROL_REG, 0xFFFF_FFFF); // Flush the current FIFO
    apbif_single_write (SEND_TO_I2S, FIFO_CONTROL_REG, 0x0000_0000); // Clear the Flush bit
    apbif_single_write (SEND_TO_I2S, MODE_REG,0x0001_0000 ); // Configure the mode register
    apbif_single_write (SEND_TO_I2S, CONTROL_REG,0x0000_0001 ); // Initiate the transfer
    wait_until_data_is_received( ); // wait for data to be transmitted
    apbif_single_write (SEND_TO_I2S, CONTROL_REG,0x0000_0000 ); // End the transfer
    apbif_multiple_read (SEND_FROM_I2S, RX_REG); // Read data which is received
};
```

27.3.4 Notes

While doing continuous change in mode such as a TX only mode / RX only mode / TX-RX mode in a single test kindly, ensure that the I2SACTIVE is only pulled low when a complete channel has been transmitted or in between the change of modes a system reset occurs.

The I2S controller should be supplied the BFS value according to the valid table even in the slave mode for correct

functioning of the I2S unit.

The flush bit for the TX and RX FIFO should be given at the start of configuration of the I2S controller and not anywhere in between for correct functioning of the device.

Always keep LRP = 1 for MSB and LSB justified mode in both Slave and Master mode.

While doing changes in TXDMAACTIVE / RXDMAACTIVE / I2SACTIVE / TXDMA PAUSE / RXDMA PAUSE / TXCHANNEL PAUSE / RXCHANNEL PAUSE / TXR in 24-bit BLC configuration, ensure that Left and Right channel data is transmitted or received successfully. Successful transmission or reception of Left and Right channel data in 24-bit BLC configuration can be observed by polling LRI bit in I2SCON register.

27.4 Register Summary

Bit	R/W	Symbol	Description	Reset Value
I2S interface control register (I2SCON) Address : C005_5000h (I2S 0) Address : C005_6000h (I2S 1) Address : C005_7000h (I2S 2)				
[31:12]	R/W	RESERVED	Reserved. Program to Zero	20h0000
[11]	R	LRI	Left / right channel clock indication: LRI depends upon LRP bit of I2SMOD register and LRCLK. 0: Left (when LRCLK is low and LRP is 0) / Right (when LRCLK is high and LRP is 0) 1: Left (when LRCLK is high and LRP is 1) / Right (when LRCLK is low and LRP is 1)	1'b1
[10]	R	FTXEMPT	Tx FIFO empty status indication 0: Tx FIFO is not empty (Ready to transmit data) 1: Tx FIFO is empty (Not ready to transmit data)	1'b1
[9]	R	FRXDEMPT	Rx FIFO empty status indication 0: Rx FIFO is not empty 1: Rx FIFO is empty	1'b1
[8]	R	FTXFULL	Tx FIFO full status indication 0: Tx FIFO is not full 1: Tx FIFO is full	1'b0
[7]	R	FRXFULL	Rx FIFO full status indication 0: Rx FIFO is not full (Ready to receive data) 1: Rx FIFO is full (Not ready to receive data)	1'b0
[6]	R/W	TXDMAPAUSE	Tx DMA operation pause command. DMA request will be halted after the current ongoing DMA transfer. 0: No pause DMA operation 1: Pause DMA operation	1'b0
[5]	R/W	RXDMAPAUSE	Rx DMA operation pause command. DMA request will be halted after the current ongoing DMA transfer. 0: No pause DMA operation 1: Pause DMA operation	1'b0
[4]	R/W	TXCHPAUSE	Tx channel operation pause command. Channel operation will be paused after LR channel data transmission is complete. 0: No pause TX channel 1: Pause TX channel	1'b0
[3]	R/W	RXCHPAUSE	RX channel operation pause command. Channel operation will be paused after LR channel data reception is complete 0: No pause RX channel 1: Pause RX channel	1'b0
[2]	R/W	TXDMACTIVE	Tx DMA Active (start DMA request) when this bit is set from high to low DMA operation will be forced to stop immediately. 0: INACTIVE 1: ACTIVE	1'b0
[1]	R/W	RXDMACTIVE	Rx DMA Active (start DMA request) when this bit is set from high to low DMA operation will be forced to stop immediately. 0: INACTIVE 1: ACTIVE	1'b0
[0]	R/W	I2SACTIVE	I2S interface active (start I2S operation command) 0: INACTIVE 1: ACTIVE	1'b0
I2C-bus Control-Status Register (ICSR) Address : C005_5004h (I2S 0) Address : C005_6004h (I2S 1) Address : C005_7004h (I2S 2)				
[31:15]	R/W	RESERVED	Reserved. Program to Zero	17h0

Bit	R/W	Symbol	Description	Reset Value
[14:13]	R/W	BLC	Bit Length Control: Bit which decides transmission of 8/16 bits per audio channel 00: 16 bits per channel 01: 8 bits per channel 10: 24 bits per channel 11: Reserved	2'b0
[12]	R/W	CDCLKCON	CODCLKOEN control 0 : CODCLKOEN is zero (I2SCODCLKO is output to the external codec) 1: CODCLKOEN is one (No clock is given to the external codec)	1'b0
[11:10]	R/W	IMS	I2S master (internal / external) or slave select mode 00: Internal master (Divide mode) : Input clock PCLK 01 : External master mode (Bypass mode) : Input clock CODCLKI 10 : Slave mode : Input clock PCLK 11 : Slave mode : Input clock CODCLKI	2'b0
[9:8]	R/W	TXR	Transmit or Receive mode select 00: Transmit only 01: Receive only 10: Transmit and Receive simultaneous mode 11: No operation / reserved	2'b0
[7]	R/W	LRP	Left / Right channel clock polarity select 0: Low for left channel and high for right channel. 1: High for left channel and low for right channel.	1'b0
[6:5]	R/W	SDF	Serial data format 00: I2S format 01: MSB-justified (Left-justified) format 10: LSB-justified (Right-justified) format 11: Reserved	2'b0
[4:3]	R/W	RFS	I2S root clock (codec clock) frequency select Fs is sampling frequency. 00: 256 fs 01: 512 fs 10: 384 fs 11: 768 fs	2'b0
[2:1]	R/W	BFS	Bit clock frequency select. Fs is sampling frequency. 00: 32 fs 01: 48 fs 10: 16 fs 11: 24 fs	2'b0
[0]	R/W	RESERVED	Reserved. Program to Zero.	1'b0
<Note> CODCLKOEN (active low) is used as a output pad enable for the I2SCODCLKO for various I2S audio codecs. Default condition of I2SMOD register has the I2SCODCLKO always enabled. Similary I2SBCLKOEN and I2SLRCLKOEN is controlled by IMS bit when 00 and 01 only then both these output enables are zero (active) which validates the output clocks at the output pads.				
I2S interface FIFO control register (I2SFIC)				
Address : C005_5008h (I2S 0)		Address : C005_6008h (I2S 1)		Address : C005_7008h (I2S 2)
[31:16]	R/W	RESERVED	Reserved. Program to Zero.	16'b0
[15]	R/W	TFULSH	Tx FIFO flush command 0: No flush 1: Flush	1'b0
[14:13]	R/W	RESERVED	Reserved. Program to Zero.	2'b0
[12:8]	R	FTXCNT	Tx FIFO data count. FIFO has 16 depth, so value ranges from 0 to 15.	5'b0

Bit	R/W	Symbol	Description	Reset Value
			N: Data count N of FIFO	
[7]	R/W	RFLUSH	Rx FIFO flush command 0: No flush 1: Flush	1'b0
[6:5]	R/W	RESERVED	Reserved. Program to Zero.	2'b0
[4:0]	R	FRXCNT	Rx FIFO data count. FIFO has 16 depth, so value ranges from 0 to 15. N: Data count N of FIFO	5'b0
I2S interface clock divider control register (I2SPSR)				
Address : C005_500Ch (I2S 0)		Address : C005_600Ch (I2S 1)		Address : C005_700Ch (I2S 2)
[31:16]	R/W	RESERVED	Reserved. Program to Zero.	16'b0
[15]	R/W	PSRAEN	Prescaler (clock divider) A active 0: Inactive 1: Active	1'b0
[14]	R/W	RESERVED	Reserved. Program to Zero.	1'b0
[13:8]	R/W	PSVALA	Prescaler (clock divider) A division value. N: Division factor is N+1	6'b0
[7:0]	R/W	RESERVED	Reserved. Program to Zero.	8'b0
I2S interface transmit data register (I2STXD)				
Address : C005_5010h (I2S 0)		Address : C005_6010h (I2S 1)		Address : C005_7010h (I2S 2)
[31:0]	W	I2STXD	Tx FIFO write data. Note that the left / right channel data is allocated as the following bit fields R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC.	32b0
I2S interface receive data register (I2SRXD)				
Address : C005_5014h (I2S 0)		Address : C005_6014h (I2S 1)		Address : C005_7014h (I2S 2)
[31:0]	R	I2SRXD	Rx FIFO read data. Note that the left / right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC.	32b0

Section 28. **AC97**

28.1 Overview

The AC97 Controller Unit in the NXP4330D/Q supports the features of AC97 revision 2.0. AC97 Controller uses audio controller link (AC-link) to communicate with AC97 Codec. Controller sends the stereo PCM data to Codec. The external digital-to-analog converter (DAC) in the Codec converts the audio sample to an analog audio waveform. Controller receives the stereo PCM data and the mono Mic data from Codec then store in memories. This Section describes the programming model for the AC97 Controller Unit. The prerequisite in this Section requires an understanding of the AC97 revision 2.0 specifications.

28.1.1 Features

The AC97 Controller includes the following features:

- Independent channels for stereo PCM In, stereo PCM Out, mono MIC In.
- DMA-based operation and interrupt based operation.
- All of the channels support only 16-bit samples.
- Variable sampling rate AC97 Codec interface (48 kHz and below)
- 16-bit, 16 entry FIFOs per channel
- Only primary Codec support

28.1.2 Block Diagram

Figure 28-1 shows the functional block diagram of NXP4330D/Q AC97 Controller. The AC97 signals from the AC-link are connected with external AC97 codec device. NXP4330D/Q AC97 Controller supports full-duplex data transfers. All digital audio streams and command/status information are communicated via AC-link.

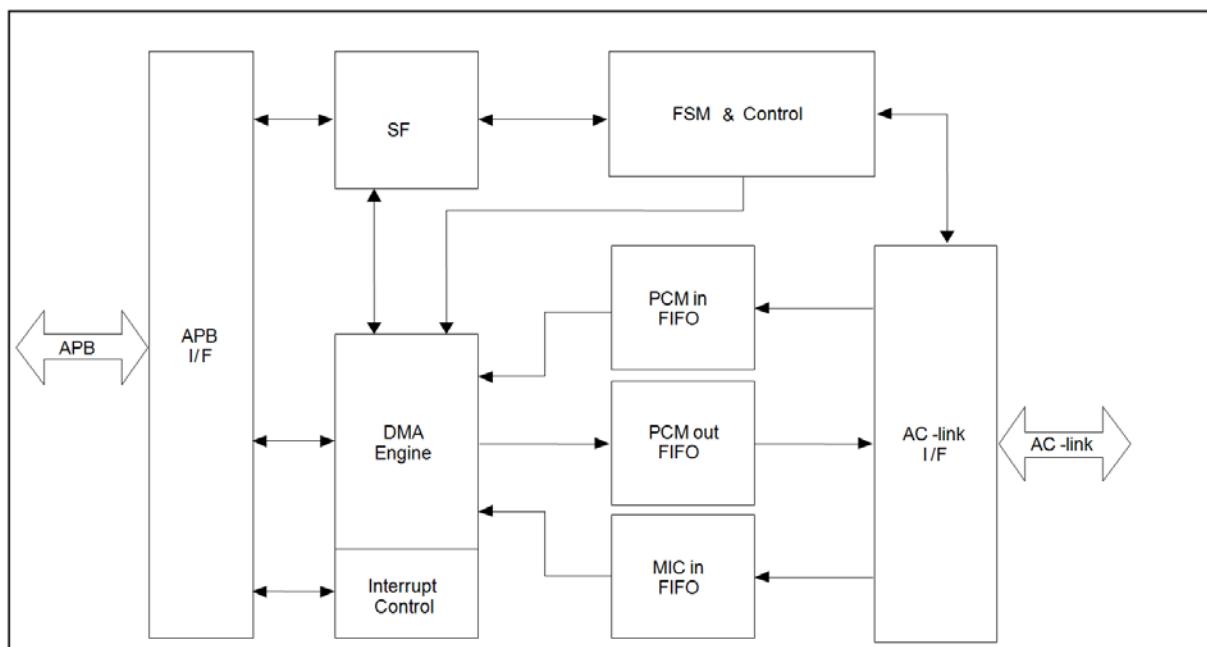


Figure 28-1. AC97 Block Diagram

28.2 Functional Description

This section explains the AC97 Controller operation, namely, AC-Link, Power-down sequence and Wake-up sequence.

28.2.1 Internal Data Path

Figure 28-2 shows the internal data path of NXP4330D/QAC97 Controller. It includes stereo Pulse Code Modulated (PCM) In, Stereo PCM Out and mono MIC-in buffers, which consist of 16-bit and 16 entries buffer. It also has 20-bit I/O shift register via AC-link.

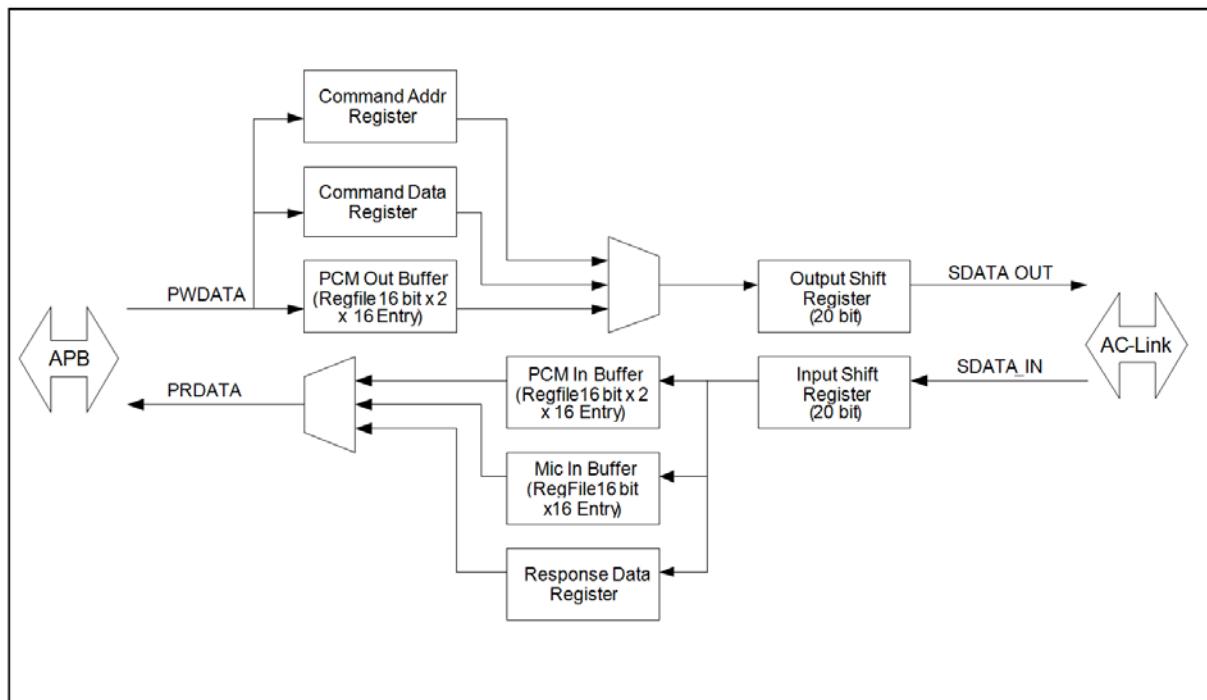


Figure 28-2. Internal Data Path

28.2.2 Operation Flow Chart

When you initialize the AC97 controller, you must assert system reset or cold reset, because the previous state of the external AC97 audio-codec is unknown. This assures that GPIO is already ready. Then you enable the codec ready interrupt. You can check codec ready interrupt by polling or interrupt. When interrupt occurs, you must de-assert codec ready interrupt. Use DMA or PIO (directly to write data to register) to transmit data from memory to register or from register to memory. If internal FIFOs (Tx FIFO or Rx FIFO) are not empty, then let data be transmitted.

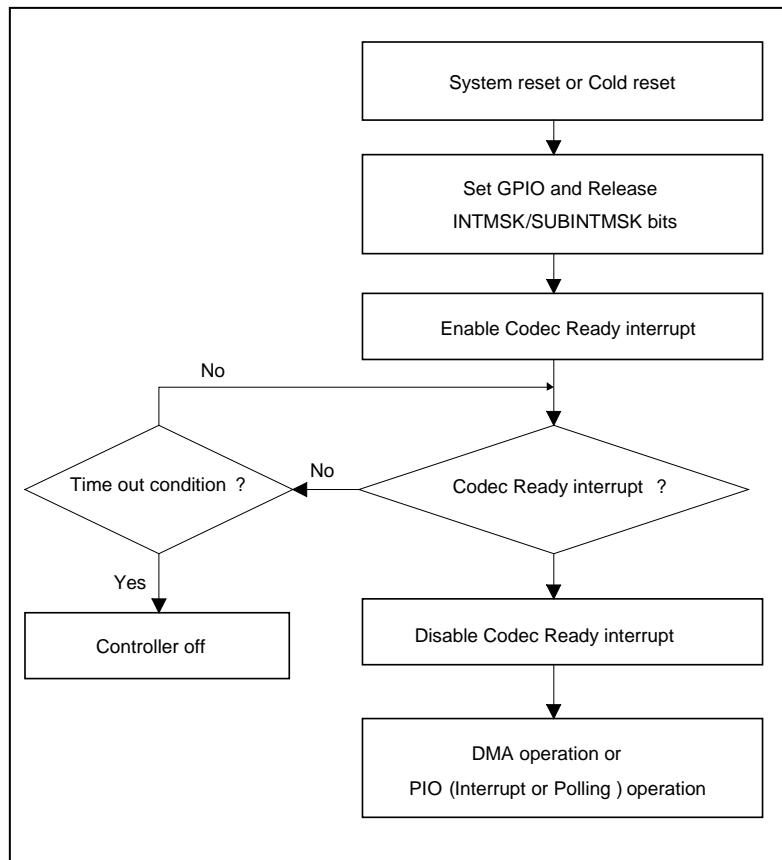


Figure 28-3. AC97 Operation Flow Chart

28.2.3 AC-link Digital Interface Protocol

Each AC97 Codec incorporates a five-pin digital serial interface that links it to the NXP4330D/QAC97 Controller. AC-link is a full-duplex, fixed-clock and PCM digital stream. It employs a time division multiplexed (TDM) scheme to handle control register accesses and multiple input and output audio streams. The AC-link architecture divides each audio frame into 12 outgoing and 12 incoming data streams. Each stream has 20-bit sample resolution and requires a DAC and an analog-to-digital converter (ADC) with a minimum 16-bit resolution.

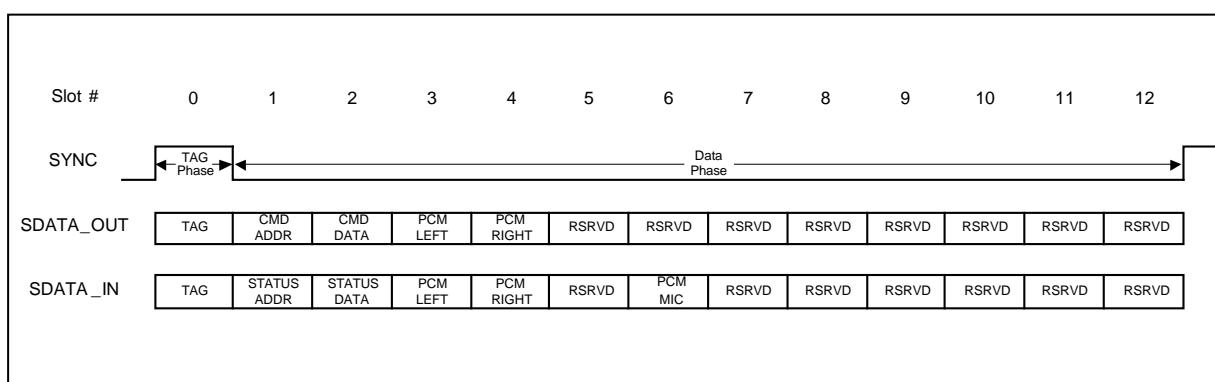


Figure 28-4. Bi-directional AC-link Frame with Slot Assignments

Figure 28-4 shows the slot definitions supported by NXP4330D/QAC97 Controller. The NXP4330D/QAC97 Controller provides synchronization for all data transaction on the AC-link.

A data transaction is made up of 256 bits of information broken into groups of 13 time slots and is called a frame. Time slot 0 is called the Tag Phase and it is 16 bits long. The other 12 time slots are called the Data Phase. The Tag Phase contains one bit that identifies a valid frame and 12 bits that identify the time slots in the Data Phase that contain valid data. Each time slot in the Data Phase is 20 bits long. A frame begins when SYNC goes high. The amount of time that SYNC is high corresponds to the Tag Phase. AC97 frames occur at fixed 48 kHz intervals and are synchronous to the 12.288 MHz bit rate clock, BITCLK.

The controller and the Codec use the SYNC and BITCLK to determine when to send transmit data and when to sample received data. A transmitter transfers the serial data stream on each rising edge of BITCLK and a receiver samples the serial data stream on falling edges of BITCLK. The transmitter must tag the valid slots in its serial data stream. The valid slots are tagged in slot 0. Serial data on the AC-link is ordered most significant bit (MSB) to least significant bit (LSB). The Tag Phase's first bit is bit[15] and the first bit of each slot in Data Phase is bit[19]. The last bit in any slot is bit[0].

28.2.3.1 AC-link Output Frame (SDATA_OUT)

Slot 0: Tag Phase

In slot 0, the first bit is a bit (SDATA_OUT, bit[15]) which represents the validity of the entire frame. If bit[15] is 1, the current frame contains at least a valid time slot. The next 12-bit positions correspond each 12 time slot contains valid data. Bits 0 and 1 of slot 0 are used as CODEC IO bits for I/O reads and writes to the CODEC registers as described in the next section. In this way, data streams of differing sample rate can be transmitted across AC-link at its fixed 48 kHz audio frame rate.

Slot 1: Command Address Port

In slot 1, it communicates control register address and write/read command information to the AC97 controller.

When software accesses the primary CODEC, the hardware configures the frame as follows:

- In slot 0, the valid bit for 1, 2 slots are set.
- In slot 1, bit[19] is set (read) or clear (write). Bits 18-12 (of slot 1) are configured to specify the index to the CODEC register. Others are filled with 0's (reserved).
- In slot 2, it configured with the data which is for writing because of output frame.

Slot 2: Command Data Port

In slot 2, this is the write data with 16-bit resolution ([19:4] is valid data)

Slot 3: PCM Playback Left channel

Slot 3 is audio output frame is the composite digital audio left stream. If a sample has a resolution that is less than 16 bits, the AC97 controller fills all training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Playback Right channel

Slot 4 which is audio output frame is the composite digital audio right stream. If a sample has a resolution that is less than 16 bits, the AC97 controller fills all training non-valid bit positions in the slot with zeroes.

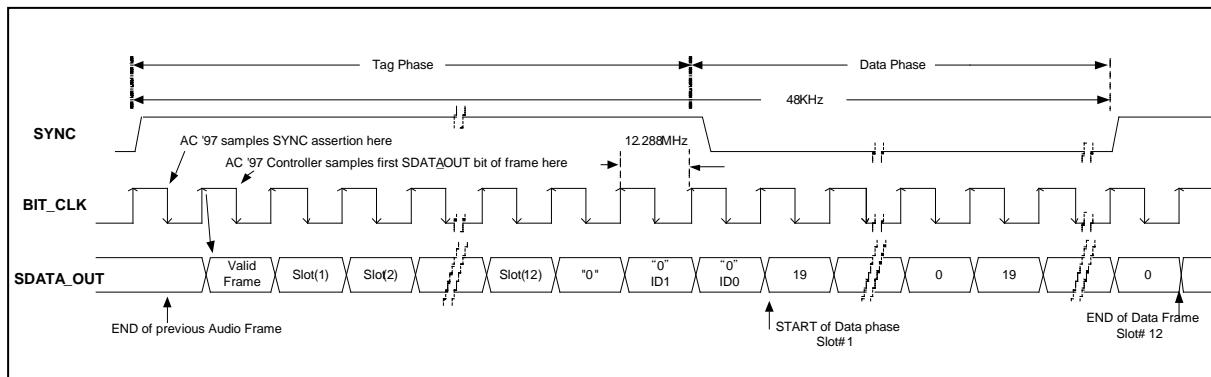


Figure 28-5. AC-link Output Frame

28.2.3.2 AC-link Input Frame (SDATA_IN)

Each AC97 Codec incorporates a five-pin digital serial interface that links it to the NXP4330D/QAC97 Controller. AC-link is a full-duplex, fixed-clock and PCM digital stream.

Slot 0: Tag Phase

In slot 0, the first bit (SDATA_OUT, bit[15]) indicates whether the AC97 controller is in the CODEC ready state. If the CODEC Ready bit is 0, it means that the AC97 controller is not ready for normal operation. This condition is normal after the power is de-asserted on reset and the AC97 controller voltage references are settling.

Slot 1: Status Address Port/SLOTREQ bits

The status port monitors the status of the AC97 controller functions. It is not limited to mixer settings and power management. Audio input frame slot 1s stream echoes the control register index for the data to be returned in slot 2, if the controller tags slots 1 and 2 as valid during slot 0. The controller only accepts status data if the accompanying status address matches the last valid command address issued during the most recent read command. For multiple sample rate output, the CODEC examines its sample-rate control registers, its FIFOs' states, and the incoming SDATA_OUT tag bits at the beginning of each audio output frame to determine which SLOTREQ bits to set active (low). SLOTREQ bits asserted during the current audio input frame indicate which output slots require data from the controller in the next audio output frame. For fixed 48 kHz operation, the SLOTREQ bits are set active (low), and a sample is transferred in each frame. For multiple sample-rate input, the "tag" bit for each input slot indicates whether valid data is present.

Bit	Description
[19]	Reserved (Filled with zero)
[18:12]	Control register index (Filled with zeroes if AC97 tags is invalid)
[11]	Slot 3 request: PCM Left channel
[10]	Slot 4 request: PCM Right channel
[9]	Slot 5 request: NA
[8]	Slot 6 request: MIC channel
[7]	Slot 7 request: NA
[6]	Slot 8 request: NA
[5]	Slot 9 request: NA
[4]	Slot 10 request: NA
[3]	Slot 11 request: NA

[2]	Slot 12 request: NA
[1:0]	Reserved (Filled with zero)

Table 28-1. Input Slot 1 Bit Definitions

Slot 2: Status Data Port

In slot 2, this is the status data with 16-bit resolution ([19:4] is valid data)

Slot 3: PCM Record Left channel

Slot 3 which is audio input frame is the left channel audio output of the AC97 Codec. If a sample has a resolution that is less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Record Right channel

Slot 4 which is audio input frame is the right channel audio output of the AC97 Codec. If a sample has a resolution that is less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 6: Microphone Record Data

The AC97 Controller supports 16-bit resolution for the MIC-in channel.

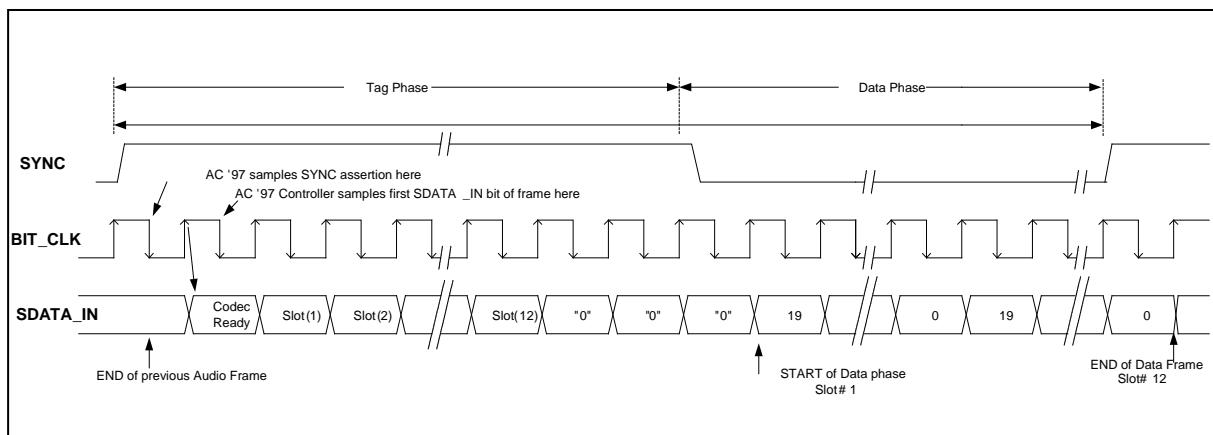


Figure 28-6. AC-Link Input Frame

28.2.4 AC97 Power-down

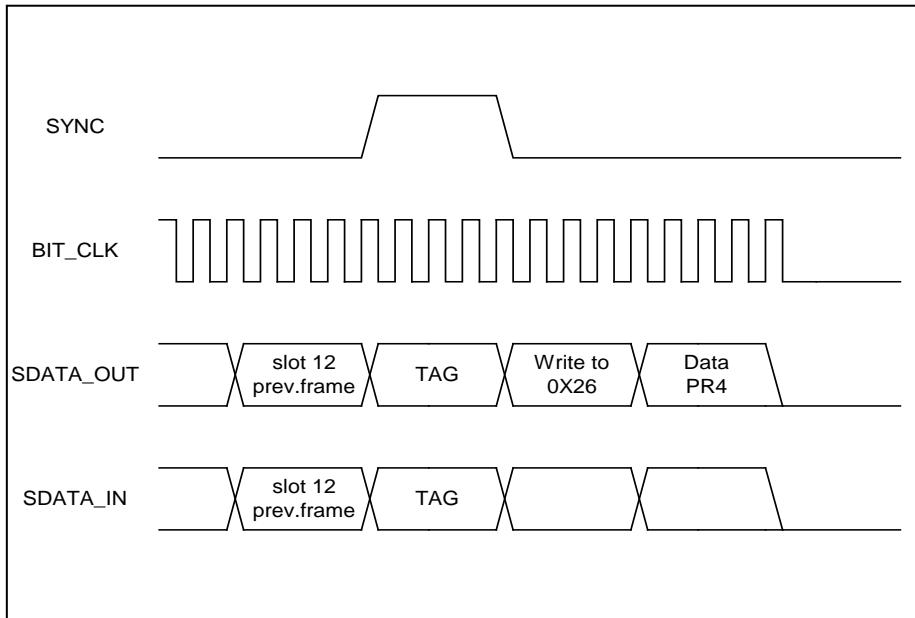


Figure 28-7. AC97 Power-Down Timing

28.2.4.1 Powering Down the AC-Link

The AC-link signals enter a low power mode when the AC97 Codec Power-down register (0x26) bit PR4 is set to 1 (by writing 0x1000). Then the Primary Codec drives both BITCLK and SDATA_IN to a logic low voltage level. The sequence follows the timing diagram as shown in Figure 28-7.

The AC97 Controller transmits the write to Power-down register (0x26) via AC-link. Set up the AC97 Controller so that it does not transmit data to slots 3-12 when it writes to the Power-down register bit PR4 (data 0x1000), and it does not require the Codec to process other data when it receives a power down request. When the Codec processes the request it immediately transfers BITCLK and SDATA_IN to a logic low level. The AC97 Controller drives SYNC and SDATA_OUT to a logic low level after programming the AC_GLBCTRL register.

28.2.4.2 Waking Up the AC-Link - Wake Up Triggered by the AC97 Controller

AC-link protocol provides a cold AC97 reset and a warm AC97 reset. The current power-down state ultimately dictates which AC97 reset is used. Registers must stay in the same state during all power-down modes unless a cold AC97 reset is performed. In a cold AC97 reset, the AC97 registers are initialized to their default values. After a power down, the AC-link must wait for a minimum of four audio frame times after the frame in which the power down occurred before it can be reactivated by reasserting the SYNC signal. When AC-link powers up, Codec ready bit (input slot 0, bit[15]) indicates that AC-link is ready for operation.

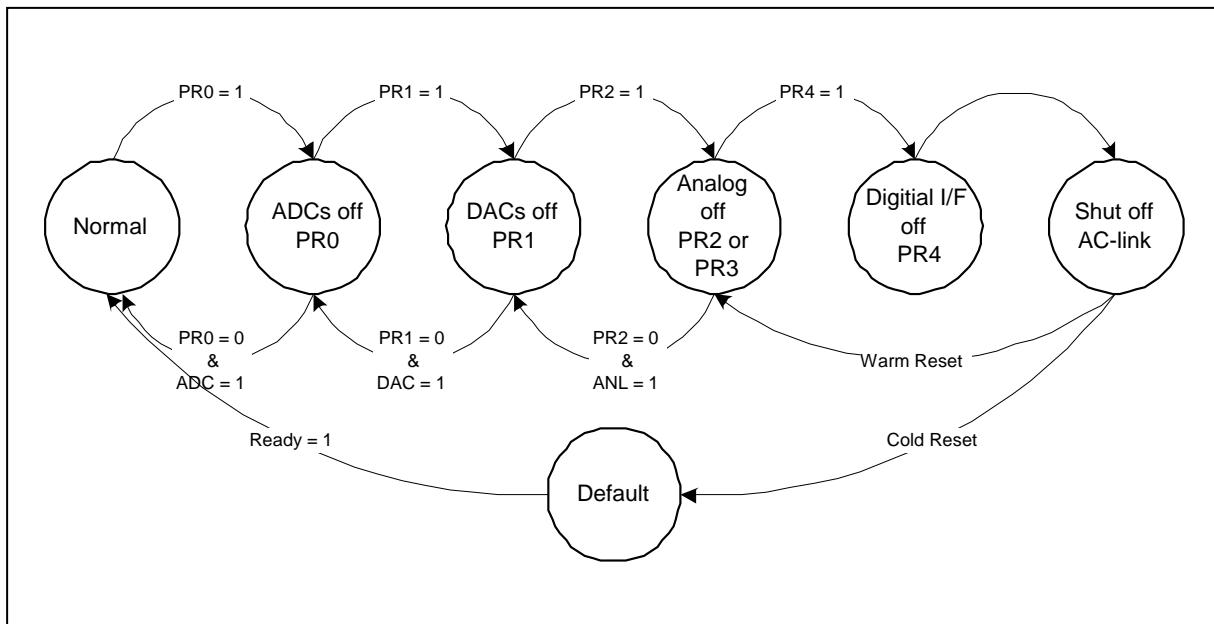


Figure 28-8. AC97 Power Down/Power Up Flow

28.2.4.3 Cold AC97 Reset

A cold reset is generated when the nRESET pin is asserted through the AC_GLBCTRL. Asserting and deasserting nRESET activates BITCLK and SDATA_OUT. A cold reset initializes all of AC97 control registers. nRESET is an asynchronous AC97 input.

28.2.4.4 Warm AC97 Reset

A Warm AC97 reset reactivates the AC-link without altering the current AC97 register values. A warm reset is generated when BITCLK is absent and SYNC is driven high. In normal audio frames, SYNC is a synchronous AC97 input. When BITCLK is absent, SYNC is treated as an asynchronous input used to generate a warm reset to AC97. The AC97 Controller must not activate BITCLK until it samples SYNC low again. This prevents a new audio frame being falsely detected.

28.2.4.5 AC97 State Diagram

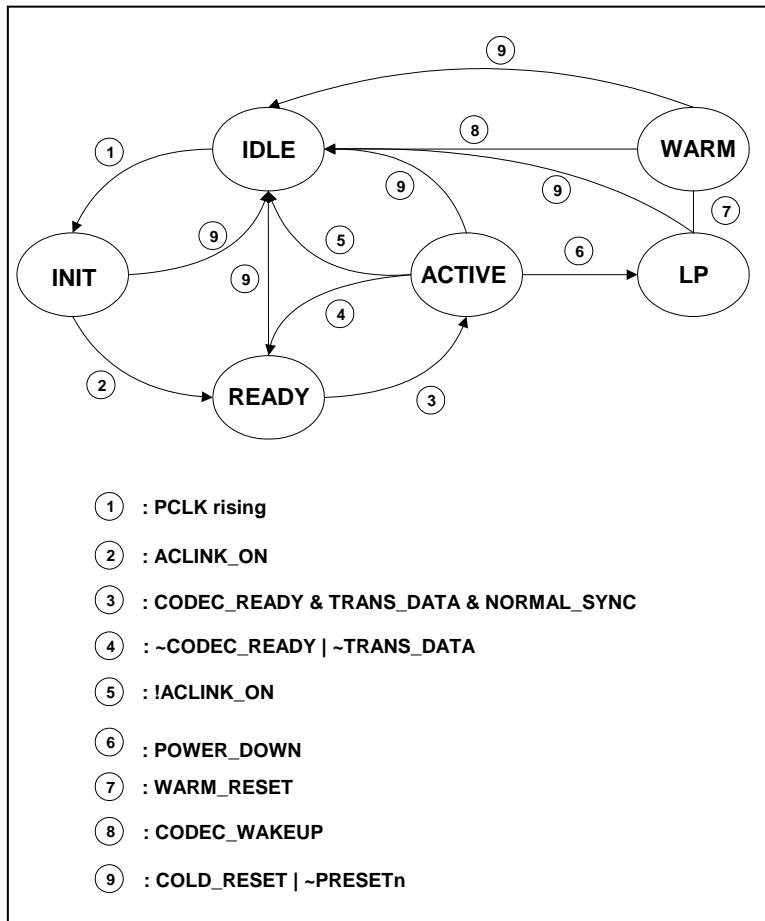


Figure 28-9. AC97 State Diagram

Figure 28-9 shows the state diagram of AC97 controller. It is useful to check AC97 controller state machine. State machine shown in above figure is synchronized by peripheral clock (PCLK). Use AC_GLBSTAT register to monitor state.

28.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
AC97 global control register(AC_GLBCTRL)				
Address : C005_8000h				
[31]	-	RESERVED	Reserved	-
[30]	R/W	CODEC READY INTERRUPT CLEAR	1 = Interrupt clear (write only)	1'b0
[29]	R/W	PCM OUT CHANNEL UNDERRUN INTERRUPT CLEAR	1 = Interrupt clear (write only)	1'b0
[28]	R/W	PCM IN CHANNEL OVERRUN INTERRUPT CLEAR	1 = Interrupt clear (write only)	1'b0
[27]	R/W	MIC IN CHANNEL OVERRUN INTERRUPT CLEAR	1 = Interrupt clear (write only)	1'b0
[26]	R/W	PCM OUT CHANNEL THRESHOLD INTERRUPT CLEAR	1 = Interrupt clear (write only)	1'b0
[25]	R/W	PCM IN CHANNEL THRESHOLD INTERRUPT CLEAR	1 = Interrupt clear (write only)	1'b0
[24]	R/W	MIC IN CHANNEL THRESHOLD INTERRUPT CLEAR	1 = Interrupt clear (write only)	1'b0
[23]	-	RESERVED	Reserved	-
[22]	R/W	CODEC READY INTERRUPT ENABLE	0 = Disables 1 = Enables	1'b0
[21]	R/W	PCM OUT CHANNEL UNDERRUN INTERRUPT ENABLE	0 = Disables 1 = Enables (FIFO is empty)	1'b0
[20]	R/W	PCM IN CHANNEL OVERRUN INTERRUPT ENABLE	0 = Disables 1 = Enables (FIFO is full)	1'b0
[19]	R/W	MIC IN CHANNEL OVERRUN INTERRUPT ENABLE	0 = Disables 1 = Enables (FIFO is full)	1'b0
[18]	R/W	PCM OUT CHANNEL THRESHOLD INTERRUPT ENABLE	0 = Disables 1 = Enables (FIFO is half empty)	1'b0
[17]	R/W	PCM IN CHANNEL THRESHOLD INTERRUPT ENABLE	0 = Disables 1 = Enables (FIFO is half full)	1'b0
[16]	R/W	MIC IN CHANNEL THRESHOLD INTERRUPT ENABLE	0 = Disables 1 = Enables (FIFO is half full)	1'b0
[15:14]	-	RESERVED	Reserved	-
[13:12]	R/W	PCM OUT CHANNEL TRANSFER MODE	00 = Off 01 = PIO 10 = DMA 11 = Reserved	2'b0
[11:10]	R/W	PCM IN CHANNEL TRANSFER MODE	00 = Off 01 = PIO	2'b0

Bit	R/W	Symbol	Description	Reset Value
			10 = DMA 11 = Reserved	
[9:8]	R/W	MIC IN CHANNEL TRANSFER MODE	00 = Off 01 = PIO 10 = DMA 11 = Reserved	2'b0
[7:4]	-	RESERVED	Reserved	-
[3]	R/W	TRANSFER DATA ENABLE USING AC-LINK	0 = Disables 1 = Enables	1'b0
[2]	R/W	AC-LINK ON	0 = Off 1 = SYNC signal transfer to Codec	1'b0
[1]	R/W	WARM RESET	0 = Normal 1 = Wake up codec from power down	1'b0
[0]	R/W	COLD RESET	0 = Normal 1 = Reset Codec and Controller logic NOTE: 1. During Cold reset, writing to any AC97 Registers is not affected. 2. When recovering from Cold reset, writing to any AC97 Registers is not affected. Example: For consecutive Cold reset and Warm reset, first set AC_GLBCTRL = 0x1 then set AC_GLBCTRL = 0x0. After recovering from cold reset set AC_GLBCTRL = 0x2 then AC_GLBCTRL = 0x0.	1'b0
AC97 global status register(AC_GLBSTAT)				
Address : C005_8004h				
[31:23]	-	RESERVED	Reserved	-
[22]	R	CODEC READY INTERRUPT	0 = Not requested 1 = Requested	1'b0
[21]	R	PCM OUT CHANNEL UNDERRUN INTERRUPT	0 = Not requested 1 = Requested	1'b0
[20]	R	PCM IN CHANNEL OVERRUN INTERRUPT	0 = Not requested 1 = Requested	1'b0
[19]	R	MIC IN CHANNEL OVERRUN INTERRUPT	0 = Not requested 1 = Requested	1'b0
[18]	R	PCM OUT CHANNEL THRESHOLD INTERRUPT	0 = Not requested 1 = Requested	1'b0
[17]	R	PCM IN CHANNEL THRESHOLD INTERRUPT	0 = Not requested 1 = Requested	1'b0
[16]	R	MIC IN CHANNEL THRESHOLD INTERRUPT	0 = Not requested 1 = Requested	1'b0
[15:3]	-	RESERVED	Reserved	-
[2:0]	R	CONTROLLER MAIN STATE	000 = Idle 001 = Init 010 = Ready	3'b0

Bit	R/W	Symbol	Description	Reset Value
			011 = Active 100 = LP 101 = Warm	
AC97 codec command register (AC_CODEC_CMD)				
<i>Address : C005_8008h</i>				
[31:24]	-	RESERVED	Reserved	-
[23]	R/W	READ ENABLE	0 = Command write 1 = Status read	1'b0
[22:16]	R/W	ADDRESS	Codec command address	7b0
[15:0]	R/W	DATA	Codec command data	16b0
AC97 codec status register (AC_CODEC_STAT)				
<i>Address : C005_800Ch</i>				
[31:23]	-	RESERVED	Reserved	-
[22:16]	R	ADDRESS	Codec command address	7b0
[15:0]	R	DATA	Codec command data	16b0
AC97 PCM out/in channel FIFO address register(AC_PCMADDR)				
<i>Address : C005_8010h</i>				
[31:28]	-	RESERVED	Reserved	-
[27:24]	R	OUT READ ADDRESS	PCM out channel FIFO read address	4'b0
[23:20]	-	RESERVED	Reserved	-
[19:16]	R	IN READ ADDRESS	PCM in channel FIFO read address	4'b0
[15:12]	-	RESERVED	Reserved	-
[11:8]	R	OUT WRITE ADDRESS	PCM out channel FIFO write address	4'b0
[7:4]	-	RESERVED	Reserved	-
[3:0]	R	IN WRITE ADDRESS	PCM in channel FIFO write address	4'b0
AC97 MIC In channel FIFO address register(AC_MICADDR)				
<i>Address : C005_8014h</i>				
[31:20]	-	RESERVED	Reserved	-
[19:16]	R	OUT WRITE ADDRESS	MIC in channel FIFO read address	4'b0
[15:4]	-	RESERVED	Reserved	-
[3:0]	R	IN WRITE ADDRESS	MIC in channel FIFO write address	4'b0
AC97 PCM out/in channel FIFO data register (AC_PCMDATA)				
<i>Address : C005_8018h</i>				
[31:16]	R/W	RIGHT DATA	PCM out/in right channel FIFO data Read = PCM in right channel Write = PCM out right channel	16'b0
[15:0]	R/W	LEFT DATA	PCM out/in left channel FIFO data Read = PCM in left channel Write = PCM out left channel	16'b0

Bit	R/W	Symbol	Description	Reset Value
AC97 MIC in channel FIFO data register(AC_MICDATA)				
Address : C005_801Ch				
[31:16]	-	RESERVED	Reserved	-
[15:0]	R/W	MONO DATA	MIC in mono channel FIFO data	16'b0

Section 29. **SPDIF TX**

29.1 Overview

The SPDIF transmitter is based on IEC60958. This Section describes a serial, uni-directional, self-clocking interface to interconnect digital audio equipment in consumer and professional applications.

When you use a consumer digital processing environment in SPDIF standard, the SPDIF interface is primarily intended to carry stereophonic programs, with a resolution of up to 20 bits per sample, an extension to 24 bits per sample being possible.

When you use SPDIF in a broadcasting studio environment, the interface is primarily intended to carry monophonic or stereophonic programs, at a 48 kHz sampling frequency and with a resolution of up to 24 bits per sample; it can carry one or two signals sampled at 32 kHz.

In both cases, the clock references and auxiliary information are transmitted with the program. IEC60958 enables the interface to carry software related data.

29.1.1 Features

Features of SPDIF-TX are:

- SPDIFOUT module only supports the consumer application in NXP4330D/Q
- Supports linear PCM up to 24-bit per sample
- Supports Non-linear PCM formats such as AC3, MPEG1 and MPEG2
- 2 x 24-bit buffers which is alternately filled with data

29.1.2 Block Diagram

Figure 29-1 illustrates the block diagram of SPDIFOUT.

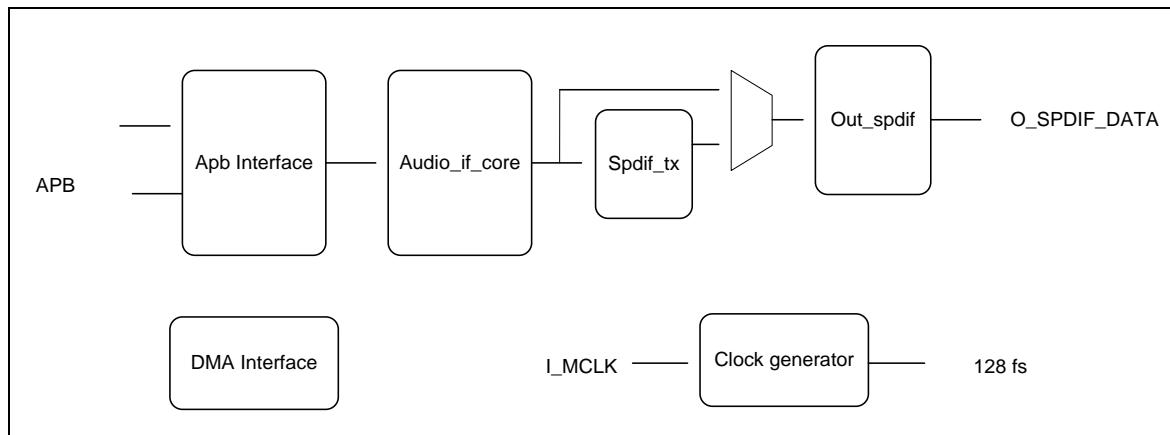


Figure 29-1. Block Diagram of SPDIFOUT

Components in SPDIF Transmitter:

- APB interface block: This block defines register banks to control the driving of SPDIFOUT module and data buffers to store linear or non-linear PCM data.
- DMA interface block: This block requests DMA service to IODMA that depends on the status of data buffer in APB Interface block.

- Clock Generator block: This block generates 128 fs (sampling frequency) clock that is used in out_spdif block from system audio clock (MCLK).
- Audio_if_core block: This block acts as an interface block between data buffer and out_spdif block. Finite-state machine controls the flow of PCM data.
- spdif_tx block: This block inserts burst preamble and executes zero-stuffing in the nonlinear PCM stream. The spdif_tx module bypasses the linear PCM data.
- out_spdif block: This block generates SPDIF format. It inserts 4-bit preamble, 16- or 20- or 24-bit data, user-data bit, validity bit, channel status bit, and parity bit into the appropriate position of 32-bit word. It modulates each bit to bi-phase format.

29.2 Functional Description

This section includes:

- Data Format of SPDIF
- Channel Coding
- Preamble
- Non-Linear PCM Encoded Source (IEC 61937)
- SPDIF Operation
- Shadowed Register

29.2.1 Data Format of SPDIF

This section includes:

- Frame Format
- Sub-frame Format (IEC 60958)

29.2.1.1 Frame Format

A frame is uniquely composed of two sub-frames. The transmission rate of frames corresponds exactly to the source sampling frequency.

In the 2-channel operation mode, the time multiplexing transmits samples taken from both channels in consecutive sub-frames.

The sub-frames related to channel 1 (left or "A" channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B after every 192 frame. This unit, which is composed of 192 frames, defines the block structure that is used to organize the channel status information. Sub-frames of channel 2 (right or "B" in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

In the single-channel operation mode in broadcasting studio environment, the frame format is identical to the 2-channel mode. Data is carried only in channel 1. In the sub-frames allocated to channel 2, time slot 28 (validity flag) should be set to logical "1" ("1" means not valid).

Figure 29-2 illustrates the format of SPDIF frame.

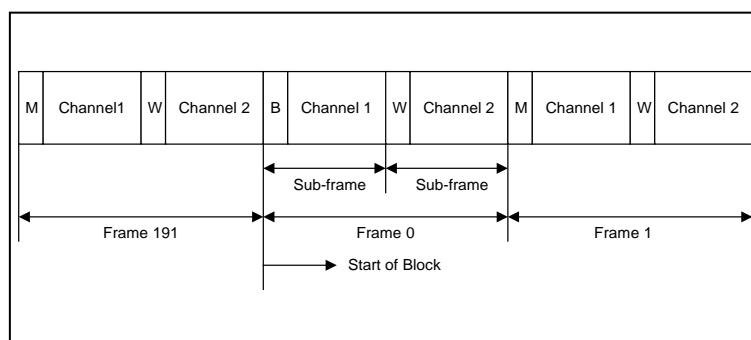


Figure 29-2. SPDIF Frame Format

29.2.1.2 Sub-frame Format (IEC 60958)

Each sub-frame is divided into 32 time slot, numbered from 0 to 31. Time slots 0 to 3 carry one of the three permitted preambles. These are used to affect synchronization of sub-frames, frames, and blocks. Time slots 4 to 27 carry the audio sample word in linear 2's complement representation. Time slot 27 carries the most significant bit... When a 24-bit coding range is used, the least significant bit is in time slot 4.

When a 20-bit coding range is sufficient, the least significant bit is in time slot 8. Time slots 4 to 7 may be used for other application. Under these circumstances, the bits in the time slots 4 to 7 are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (20 or 24 bits), the unused least significant bits shall be set to a logical "0". This procedure supports that SPDIF connect equipment by using different numbers of bits. Time slot 28 carries the validity flag associated with the audio sample word. This flag is set to logical "0" when the audio sample is reliable. Time slot 29 carries 1 bit of the user data associated with the audio channel that is transmitted in the same sub-frame. The default value of the user bit is logical "0".

Time slot 30 carries 1 bit of the channel status words associated with the audio channel transmitted in the same sub-frame. Time slot 31 carries a parity bit such that the time slots, including 4 to 31 carries an even number of ones and zeros.

Figure 29-3 illustrates format of SPDIF sub-frame.

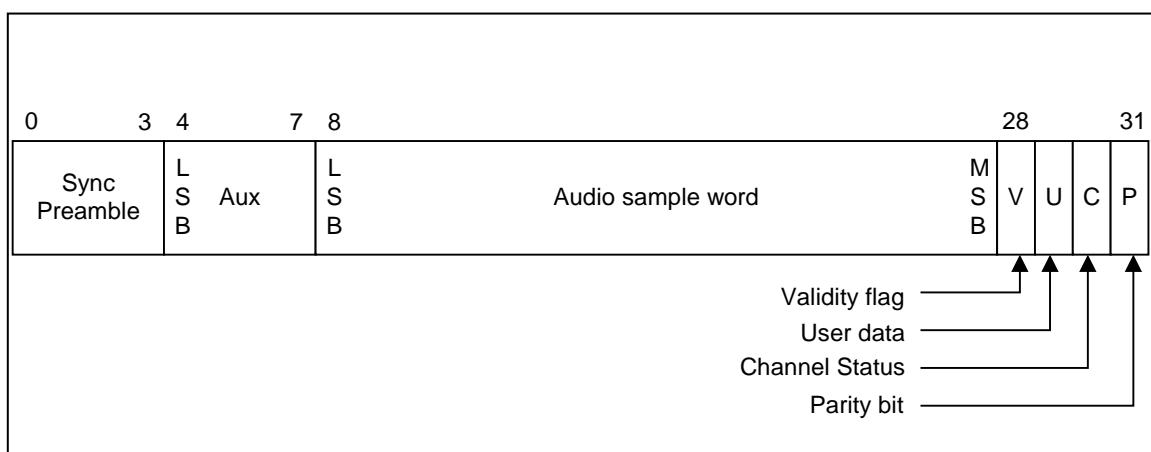


Figure 29-3. SPDIF Sub-frame Format

29.2.2 Channel Coding

Time slots 4 to 31 are encoded in biphasemark to:

- Minimize the dc component on the transmission line
- Facilitate clock recovery from the data stream
- Make the interface insensitive to the polarity of connections

A symbol comprising two consecutive binary states represents each bit to be transmitted:

- The first state of a symbol is always different from the second state of the previous symbol.
- The second state of the symbol is identical to the first when the bit to be transmitted is logical "0" and is different from the first when the bit is logical "1".

Figure 29-4 illustrates channel coding.

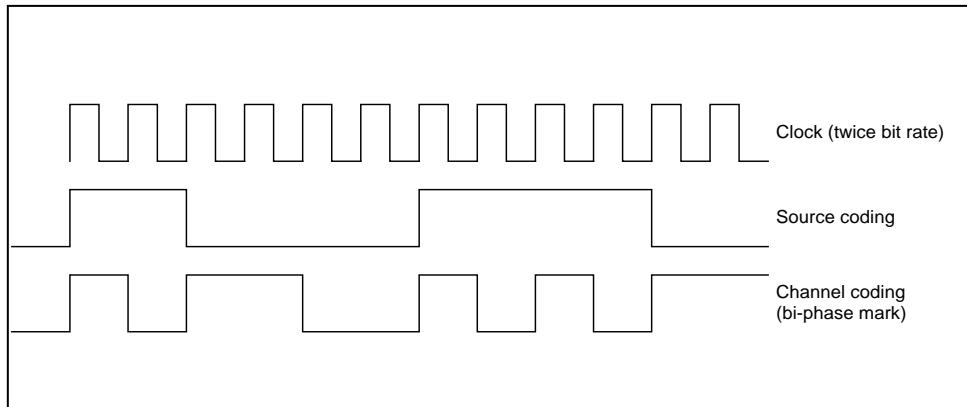


Figure 29-4. SPDIF Sub-frame Format

29.2.3 Preamble

Preambles are specific patterns that provide synchronization and identification of the sub-frames and blocks. A set of three preambles (M, B, and W) is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.

Similar to bi-phase code, these preambles are dc free and provide clock recovery. They differ in minimum two states from any valid bi-phase sequence.

29.2.4 Non-Linear PCM Encoded Source (IEC 61937)

The non-linear PCM encoded audio bit stream is transferred by using the basic 16-bit data area of the IEC 60958 sub frames, that is, in time slots 12 to 27. Each IEC 60958 frame can transfer 32 bits of the non-PCM data in consumer application mode.

When the SPDIF bit stream conveys linear PCM audio, the symbol frequency is 64 times the PCM sampling frequency (32 time slots per two channels of PCM sample time). When a non-linear PCM encoded audio bit stream is transmitted by the interface, the symbol frequency shall be 64 times the sampling rate of the encoded audio within that bit stream. If a non-linear PCM encoded audio bit stream is transmitted by the interface containing audio with low sampling frequency, the symbol frequency shall be 128 times the sampling rate of the encoded audio within that bit stream.

Each data burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc, and Pd), followed by the burst-payload, which contains data of an encoded audio frame.

The burst-preamble consists of four mandatory fields:

- Pa and Pb represent a synchronization word.
- Pc provides information about the type of data and some information/control for the receiver.
- Pd provides the length of the burst-payload, limited to 216 (= 65,535) bits.

The four preamble words are contained in two sequential SPDIF frames. The frame beginning the data-burst contains preamble word Pa in sub-frame 1 and Pb in sub-frame 2. The next frame contains Pc in sub-frame 1 and Pd in sub-frame 2. When placed into a SPDIF sub-frame, the MSB of a 16-bit burst-preamble is placed into time slot 27 and the LSB is placed into time slot 12.

Figure 29-5 illustrates format of Burst Payload.

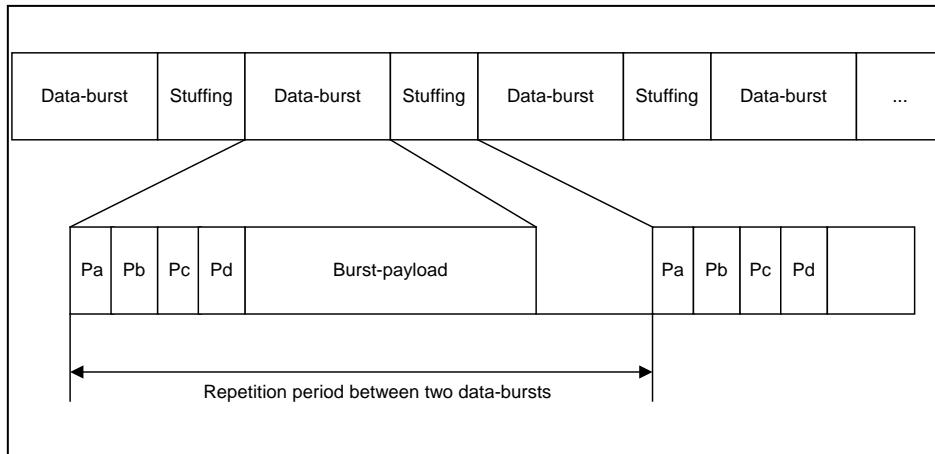


Figure 29-5. Format of Burst Payload

Table 29-1 lists the burst preamble words.

Preamble Word	Length of Field	Contents	Value MSB LSB
Pa	16 bits	Sync word 1	0xF872
Pb	16 bits	Sync word 2	0x4E1F
Pc	16 bits	Burst-info	Refer to SPDBSTAS_SHD[15:0] for more information.
Pd	16 bits	Length-code	Refer to SPDBSTAS_SHD[31:16] for more information.

Table 29-1. Burst Preamble Words

29.2.5 SPDIF Operation

The bit frequency of SPDIF is 128 fs (fs: sampling frequency). Therefore, divide audio main clock (MCLK) depending on the frequency of MCLK to make the main clock of SPDIF. You can divide MCLK by:

- 2 in case of 256 fs
- 3 in case of 384 fs
- 4 in case of 512 fs

The SPDIF module in Exynos5250 changes the audio sample data format to SPDIF. To change the format, SPDIF module inserts these into the appropriate time slots:

- Preamble data
- Channel status data
- User data
- Error check bit
- Parity bit

Preamble data are fixed in the module and inserted depending on sub-frame counter. Channel status data are set in the SPDCSTAS register and used by 1 bit per frame. User data always have zero values.

For non-linear PCM data, insert burst-preamble, which consists of Pa, Pb, Pc, and Pd before burst-payload and zero is padded from the end of burst-payload to the repetition count. Pa (= 16'hF872) and Pb (= 16'h4E1F) is fixed in the module and Pc and Pd is set in the register SPDBSTAS. To stuff zero, the end of burst-payload is calculated from Pd value and repetition count that depends on data type in the preamble Pc is acquired from register SPDCNT.

Audio data are justified to the LSB. 16-, 20- or 24-bit PCM data and 16-bit stream data are supported. The unoccupied

upper bits of 32-bit word are ignored.

Data are fetched via DMA request. When one of two data buffers is empty, DMA service is requested. Audio data that are stored in the data buffers are transformed into SPDIF format and output to the port. For non-linear PCM data, interrupt is generated after audio data are output up to the value specified in the SPDCNT register. Interrupt sets the registers such as SPDBSTAS and SPDCNT to new values when the data type of new bit stream is different from the previous one.

29.2.6 Shadowed Register

Both SPDBSTAS_SHD and SPDCNT_SHD registers are shadowed registers that are related to SPDBSTAS and SPDCNT registers, respectively. They are updated from related registers at every stream end interrupt signal. The usage of shadowed register is as follows.

- 1) Set burst status and repetition count information to their respective registers.
- 2) Turn on SPDIF module, and stream end interrupt is asserted immediately.
- 3) With stream end interrupt, shadowed registers are updated from their related registers and SPDIF starts to transfer data.
- 4) The next stream information (burst status and repetition count) can be written to SPDBSTAS and SPDCNT register because the previous information is copied to their respective shadowed registers.
- 5) Set next stream information to SPDBSTAS and SPDCNT registers.
- 6) Wait for stream end interrupt that signals the end of the first stream.
- 7) With stream end interrupt, the second stream data will start to transfer.
- 8) Set third stream information to registers.

The usage of user bit registers is similar to stream information registers. However, these registers are not related to SPDIF end interrupt but to user data interrupt. As soon as SPDIF is on, the shadowed user bit registers are updated from their related registers and user bit starts to be shifted out with asserted user data interrupt. User can write the next user data to registers with this interrupt. After entire 96 user bits are shifted out, user data interrupt will be asserted again and 3rd user bits can be written to registers with second user bits going out.

29.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
Clock control register(SPDCLKCON)				
Address : C005_9000h				
[31:4]	-	RESERVED	Reserved	-
[3:2]	R/W	MCLK SEL	Main audio clock selection 0 = Internal Clock (I_MCLK_INT)1 = External Clock (I_MCLK_EXT0)	2'h0
[1]	R	SPDIFOUT CLOCK DOWN READY	0 = Clock-down not ready 1 = Clock-down ready	1'h0
[0]	R/W	SPDIFOUT POWER ON	0 = Power off1 = Power on	1'h0
Control register(SPDCON)				
Address : C005_9004h				
[31]	-	RESERVED	Reserved	-
[26:22]	R	FIFO LEVEL	FIFO Level Monitoring (Read Only) FIFO depth is 16 0 = Empty of FIFO Level16 = Full of FIFO Level	5'h0
[21:19]	R/W	FIFO LEVEL THRESHOLD	FIFO Threshold Level is controllable 000 = 0-FIFO Level001 = 1-FIFO Level 010 = 4-FIFO Level011 = 6-FIFO Level 100 = 10-FIFO Level101 = 12-FIFO Level 110 = 14-FIFO Level111 = 15-FIFO Level	3'h0
[18:17]	R/W	FIFO TRANSFER MODE	00 = DMA transfer mode01 = Polling mode 10 = Interrupt mode11 = Reserved	2'h0
[16]	R/W	FIFO_LEVEL INTERRUPT STATUS	Read Operation 0 = No interrupt pending1 = Interrupt pending Write Operation 0 = No effect1 = Clear this flag	1'h0
[15]	R/W	FIFO_LEVEL INTERRUPT ENABLE	0 = Interrupt masked 1 = Enable interrupt	1'h0
[14:13]	R/W	ENDIAN FORMAT	00 = big endian o_data = {in_data[23:0]}; 01 = 4 byte swap o_data={in_data[15:8], in_data[23:16], in_data[31:24]}; 10 = 3 byte swap o_data={in_data[7:0], in_data[15:8], in_data[23:16]}; 11 = 2 byte swap o_data={0x00,in_data[7:0], in_data[15:8]}	2'h0

Bit	R/W	Symbol	Description	Reset Value
			*in_data: BUS ->in port of SPDIF o_data: in port of SPDIF -> Logic	
[12]	R/W	USER_DATA_ATTACH	0 = User data is stored in USERBIT register. User data of sub-frame is out from USERBIT1, 2, 3 (96-bit) 1 = User data is stored in 23rd bit of audio data. User data is out in 23th bit of PCM data.	1'h0
[11]	R/W	USER DATA INTERRUPT STATUS	Read Operation 0 = No interrupt pending 1 = Interrupt pending when 96-bit of user data is out. Write Operation 0 = No effect 1 = Clear this flag	1'h0
[10]	R/W	USER DATA INTERRUPT ENABLE	0 = Interrupt masked 1 = Enables interrupt	1'h0
[9]	R/W	BUFFER EMPTY INTERRUPT STATUS	Read Operation 0 = No interrupt pending 1 = Interrupt pending Write Operation 0 = No effect 1 = Clears this flag	1'h0
[8]	R/W	BUFFER EMPTY INTERRUPT ENABLE	0 = Interrupt masked 1 = Enables interrupt	1'h0
[7]	R/W	STREAM END INTERRUPT STATUS	Read Operation 0 = No interrupt pending 1 = Interrupt pending when the number of output audio data reaches repetition count in SPDCNT register Write Operation 0 = No effect 1 = Clears this flag.	1'h0
[6]	R/W	STREAM END INTERRUPT ENABLE	0 = Interrupt masked 1 = Enables interrupt	1'h0
[5]	R/W	SOFTWARE RESET	0 = Normal operation 1 = Software reset Software reset is 1-cycle pulse (auto clear) Enables I_MCLK before software reset assertion because SPDIF uses synchronous reset	1'h0
[4:3]	R/W	MAIN AUDIO CLOCK FREQUENCY	00 = 256 fs 01 = 384 fs 10 = 512 fs 11 = Reserved If you want to use SPDIF on HDMI, select 512 fs because HDMI in Exynos5250 accepts only 512 fs or more frequency.	2'h0
[2:1]	R/W	PCM DATA SIZE	00 = 16 bits 01 = 20 bits 10 = 24 bit 11 = Reserved	2'h0
[0]	R/W	PCM OR STREAM	0 = Stream 1 = PCM	1'h0
Burst status register(SPDBSTAS)				
<i>Address : C005_9008h</i>				
[31:16]	R/W	BURST DATA LENGTH BIT	ES size in bits (Burst Preamble Pd) ES size: Elementary Stream size This indicates Burst-payload length	16'h0
[15:13]	R/W	BIT STREAM NUMBER	Bit_stream_number should be set to 0	3'h0

Bit	R/W	Symbol	Description	Reset Value
[12:8]	R/W	DATA TYPE DEPENDENT INFO	Data type dependent information	5'h0
[7]	R/W	ERROR FLAG	0 = Error flag indicates a valid burst_payload 1 = Error flag indicates that the burst payload may contain errors	1'h0
[6:5]	-	RESERVED	Reserved	-
[4:0]	R/W	COMPRESSED DATA TYPE	00000 = Null Data 00001 = AC-3 00010 = Reserved 00011 = Pause 00100 = MPEG1 (layer1) 00101 = MPEG1 (layer2, 3), MPEG2-bc 00110 = MPEG2 ? extension 00111 = Reserved 01000 = MPEG2 (layer1 ? lsf) 01001 = MPEG2 (layer2, layer3 ? lsf) Others = Reserved	5'h0
Channel status register(SPDCSTAS)				
Address : C005_900Ch				
[31:30]	-	RESERVED	Reserved	-
[29:28]	R/W	CLOCK ACCURACY	10 = Level I, ± 50 ppm 00 = Level II, ± 1000 ppm 01 = Level III, variable pitch shifted	2'h0
[27:24]	R/W	SAMPLING FREQUENCY	0000 = 44.1 kHz 0010 = 48 kHz 0011 = 32 kHz 1010 = 96 kHz	4'h0
[23:20]	R/W	CHANNEL NUMBER	Bit[20] is LSB	4'h0
[19:16]	R/W	SOURCE NUMBER	Bit[16] is LSB	4'h0
[15:8]	R/W	CATEGORY CODE	Equipment type CD player = 0000_0001 DAT player = L000_0011 DCC player = L100_0011 Mini disc = L100_1001 (L: Information about generation status of the material)	8'h0
[7:6]	R/W	CHANNEL STATUS MODE	00 = Mode 0 Others = Reserved	2'h0
[5:3]	R/W	EMPHASIS	When bit[1] = 0, 000 = 2 Audio channels without pre-emphasis 001 = 2 Audio channels with 50 us / 15 us pre-emphasis When bit[1] = 1, 000 = Default state	3'h0

Bit	R/W	Symbol	Description	Reset Value
[2]	R/W	COPYRIGHT ASSERTION	0=Copyright 1=No copyright	1'h0
[1]	R/W	AUDIO SAMPLE WORD	0=Linear PCM 1=Non-linear PCM	1'h0
[0]	R/W	CHANNEL STATUS BLOCK	0=Consumer format 1=Professional format	1'h0
SPDIFOUT data buffer(SPDDAT)				
<i>Address : C005_9010h</i>				
[31:24]	-	RESERVED	Reserved	-
[23:0]	W	SPDIFOUT DATA	PCM or stream data	24'h0
Repetition count register(SPDCNT)				
<i>Address : C005_9014h</i>				
[31:13]	-	RESERVED	Reserved	-
[12:0]	W	STREAM REPETITION COUNT	Repetition count according to data type. This bit is valid only for stream data.	13'h0
Shadowed burst status register(SPDBSTAS_SHD)				
<i>Address : C005_9018h</i>				
[31:16]	R	BURST DATA LENGTH BIT	ES size in bits (Burst Preamble Pd) ES size: Elementary Stream size This indicates Burst_payload length	16'h0
[15:13]	R	BIT STREAM NUMBER	Bit_stream_number should be set to 0	3'h0
[12:8]	R	DATA TYPE DEPENDENT INFO	Data type dependent information	5'h0
[7]	R	ERROR FLAG	0 = Error flag indicates a valid burst_payload 1 = Error flag indicates that the burst payload may contain errors	1'h0
[6:5]	-	RESERVED	Reserved	-
[4:0]	R	COMPRESSED DATA TYPE	00000 = Null Data 00001 = AC-3 00010 = Reserved 00011 = Pause 00100 = MPEG1 (layer1) 00101 = MPEG1 (layer2, 3), MPEG2-bc 00110 = MPEG2 ? extension 00111 = Reserved 01000 = MPEG2 (layer1 ? lsf) 01001 = MPEG2 (layer2, layer3 ? lsf) Others = Reserved	5'h0
Shadowed repetition count register(SPDCNT_SHD)				
<i>Address : C005_901Ch</i>				

Bit	R/W	Symbol	Description	Reset Value
[31:13]	-	RESERVED	Reserved	-
[12:0]	R	STREAM REPETITION COUNT	Repetition count according to data type This bit is valid only for stream data.	13h0
Sub-code Q1 to Q32(USERBIT1)				
<i>Address : C005_9020h</i>				
[31:0]	R/W	USER DATA BIT (SUB-CODE Q FOR CD)	USERBIT1: Q1 to Q32 User Data Bit has the Digital Audio Track information (Track number, Play Time and so on). 1176 bits of these being taken out in a row.	32h0
Sub-code Q33 to Q64(USERBIT2)				
<i>Address : C005_9024h</i>				
[31:0]	R/W	USER DATA BIT (SUB-CODE Q FOR CD)	USERBIT2: Q33 to Q64 User Data Bit has the Digital Audio Track information (Track number, Play Time and so on). 1176 bits of these being taken out in a row.	32h0
Sub-code Q65 to Q96(USERBIT3)				
<i>Address : C005_9028h</i>				
[31:0]	R/W	USER DATA BIT (SUB-CODE Q FOR CD)	USERBIT3: Q65 to Q96 User Data Bit has the Digital Audio Track information (Track number, Play Time and so on). 1176 bits of these being taken out in a row.	32h0
Shadowed register userbit1(USERBIT1_SHD)				
<i>Address : C005_902Ch</i>				
[31:0]	R	USER DATA BIT	USERBIT1_SHD: Q1 to Q32 User Data Bit has the Digital Audio Track information like (Track number, Play Time etc.). 1176 bits of these being taken out in a row.	32h0
Shadowed register userbit2(USERBIT2_SHD)				
<i>Address : C005_9030h</i>				
[31:0]	R	USER DATA BIT	USERBIT2_SHD: Q33 to Q64 User Data Bit has the Digital Audio Track information like (Track number, Play Time etc.). 1176 bits of these being taken out in a row.	32h0
Shadowed register userbit3(USERBIT3_SHD)				
<i>Address : C005_9034h</i>				
[31:0]	R	USER DATA BIT	USERBIT3_SHD: Q65 to Q96 User Data Bit has the Digital Audio Track information like (Track number, Play Time etc.). 1176 bits of these being taken out in a row.	32h0
RTL version information(VERSION_INFO)				
<i>Address : C005_9038h</i>				
[31:0]	R	VERSION INFORMATION	RTL Version Information	32hD

Section 30. **SPDIF RX**

30.1 Overview

SPDIF standard defines a serial interface for transferring digital audio data between various audio equipments like DVD/HD-DVD players, AVR's and amplifiers. When audio is transferred from a DVD player to an audio amplifier over an analogue link, noise is introduced. Filtering out this noise is a difficult task. This problem is overcome when audio data is transferred over a digital link instead of an analogue link. The data can be transferred between devices without having to convert it to an analogue signal. This is the biggest advantage of SPDIF.

30.1.1 Features

Features of SPDIF-RX are:

- Serial, unidirectional, self-clocking interface
- Single wire-single signal interface
- Easy to work because it is polarity independent

30.1.2 Block Diagram

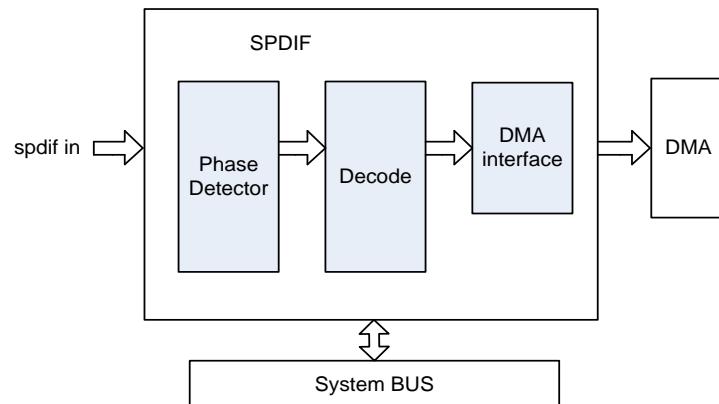


Figure 30-1. SPDIF-RX Block Diagram

30.2 Functional Description

SPDIF is a single wire serial interface and the clock is embedded within the data. The transmitted data is bi-phase mark encoded. The clock and frame sync is recovered at the receiver along with bi-phase decoded data stream. Each data bit in the stream has a time slot. The time slot begins with a transition and ends with a transition. If the transmitted data bit is "1" then additional transition is made in the middle of the time slot. Data bit "0" does not have extra transition. The shortest interval between the transitions is called the unit interval (UI).

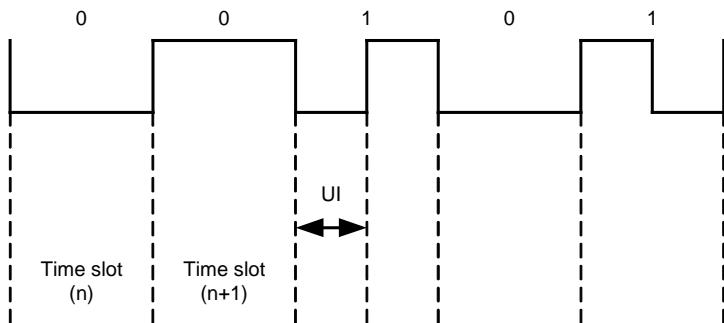


Figure 30-2. SPDIF bi-phase mark encoded stream

The least significant bit of the data is driven first. Each frame is 64 timeslots and has two sub-frames, which are 32 timeslots (Figure 30-3). The sub-frame starts with a preamble followed by 24 bits of data and ends with 4 bits which carry information such as user data and channel status. The first four time slots of a sub-frame, called preamble, is used to indicate sub-frame and block starts.

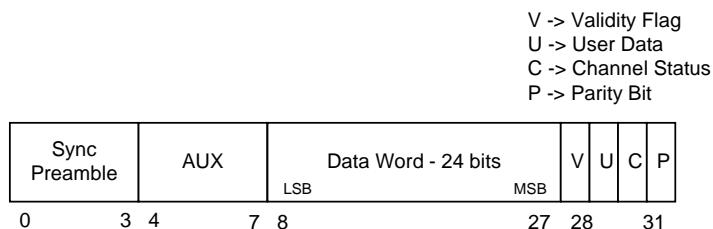


Figure 30-3. SPDIF sub-frame format

There are three preambles, each of which breaks the bi-phase coding rule by containing one or two pulses, which have duration of 3 UIs. This would mean that the pattern can't occur anywhere else in the stream. Each sub-frame begins with a 4-bit preamble. Start of a block is indicated by preamble "Z" and the start of sub-frame channel "A". Preamble "X" indicates the start of a channel "A" sub-frame when not at the start of a block while preamble "Y" indicates the start of a channel "B" sub-frame.

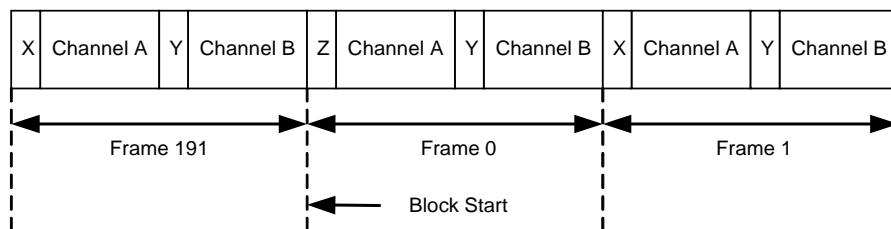


Figure 30-4. SPDIF frame and block format

30.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
SPDIF-RX Control Register (SPDIF_CTRL)				
<i>Address : C005_A000h</i>				
[31:12]	-	RESERVED	Reserved	-
[11]	R/W	FILLREGUSERINV	User Data filling order. 0 : Fill data from LSB to MSB 1 : Fill data from MSB to LSB	1'b0
[10]	R	LOCK	Lock to SPDIF input enable or disable. 0 : No-lock 1 : Lock	1'b0
[9]	R/W	CLR_FIFO	DMA write/read count clear 0 : Enable 1 : Clear	1'b0
[8]	R/W	ENBPASEDET	Specifies whether the phase detector is enable or disable. 0 : Disable 1 : Enable	1'b0
[7:4]	R/W	SAMPLE_OFFSET	Specifies the valid sampling bit. 4'b0000 : 8 4'b0010 : 6 4'b0100 : 4 4'b0110 : 2 4'b1000 : 0	4'b0001:7
[3]	R/W	ENBCAPUSERSTAT	Capture User Data. When start-of-block pulse is TRUE, RegUserA, RegUserB, RegStatA, RegStatB, detected rx_data can be captured. 0 : Disable 1 : Enable	1'b0
[2]	R/W	DMA_DATAONLY	Specifies whether DMA transfer the data only 0 : Data & Status 1 : Data only	1'b0
[1]	R/W	DMA_SWAP	Specifies the order of ChannelA and ChannelB. 0 : ChannelA first 1 : ChannelB first	1'b0
[0]	R/W	DECODE_ENB	Begin to decoder 0 : Disable 1 : Enable	1'b0
SPDIF-RX Interrupt Register (SPDIF_ENBIRQ)				
<i>Address : C005_A004h</i>				
[31:8]	-	RESERVED	Reserved	-
[7]	R/W	PENDLOCK	Interrupt pending bit of LOCK detect. Read : 0 : Not pended Write : 0 : No affect	1'b0
[6]	R/W	PENDERR	Interrupt pending bit of ERROR detect. Read : 0 : Not pended Write : 0 : No affect	1'b0
[5]	R/W	PENDPARITY	Interrupt pending bit of PARITY detect. Read : 0 : Not pended Write : 0 : No affect	1'b0

Bit	R/W	Symbol	Description	Reset Value
[4]	R/W	PENDBLOCK	Interrupt pending bit of BLOCK detect. Read : 0 : Not pended Write : 0 : No affect 1 : Pended 1 : Clear	1'b0
[3]	R/W	ENBIRQLOCK	LOCK IRQ Enable 0 : Disable 1 : Enable	1'b0
[2]	R/W	ENBIRQERR	ERROR IRQ Enable 0 : Disable 1 : Enable	1'b0
[1]	R/W	ENBIRQPARTY	PARTY IRQ Enable 0 : Disable 1 : Enable	1'b0
[0]	R/W	ENBIRQBLOCK	BLOCK IRQ Enable 0 : Disable 1 : Enable	1'b0
UserA Register[31:0] (REGUSERA0)				
<i>Address : C005_A008h</i>				
[31:0]	R	REGUSERA[31:0]	Read UserA Register[31:0] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. A enable.	32'h0
UserA Register[63:32] (REGUSERA1)				
<i>Address : C005_A00Ch</i>				
[31:0]	R	REGUSERA[63:32]	Read UserA Register[63:32] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. A enable.	32'h0
UserA Register[95:64] (REGUSERA2)				
<i>Address : C005_A010h</i>				
[31:0]	R	REGUSERA[95:64]	Read UserA Register[95:64] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. A enable.	32'h0
UserA Register[127:96] (REGUSERA3)				
<i>Address : C005_A014h</i>				
[31:0]	R	REGUSERA[127:96]	Read UserA Register[127:96] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. A enable.	32'h0
UserA Register[159:128] (REGUSERA4)				
<i>Address : C005_A018h</i>				
[31:0]	R	REGUSERA[159:128]	Read UserA Register[159:128] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. A enable.	32'h0
UserA Register[191:160] (REGUSERA5)				
<i>Address : C005_A01Ch</i>				
[31:0]	R	REGUSERA[191:160]	Read UserA Register[191:160] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. A enable.	32'h0
UserB Register[31:0] (REGUSERB0)				
<i>Address : C005_A020h</i>				
[31:0]	R	REGUSERB[31:0]	Read UserB Register[31:0] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. B enable.	32'h0
UserB Register[63:32] (REGUSERB1)				
<i>Address : C005_A024h</i>				
[31:0]	R	REGUSERB[63:32]	Read UserB Register[63:32] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. B enable.	32'h0

Bit	R/W	Symbol	Description	Reset Value
UserB Register[95:64] (REGUSERB2)				
<i>Address : C005_A028h</i>				
[31:0]	R	REGUSERB[95:64]	Read UserB Register[95:64] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. B enable.	32h0
UserB Register[127:96] (REGUSERB3)				
<i>Address : C005_A02Ch</i>				
[31:0]	R	REGUSERB[127:96]	Read UserB Register[127:96] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. B enable.	32h0
UserB Register[159:128] (REGUSERB4)				
<i>Address : C005_A030h</i>				
[31:0]	R	REGUSERB[159:128]	Read UserB Register[159:128] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. B enable.	32h0
UserB Register[191:160] (REGUSERB5)				
<i>Address : C005_A034h</i>				
[31:0]	R	REGUSERB[191:160]	Read UserB Register[191:160] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and user data ch. B enable.	32h0
StatA Register[31:0] (REGSTATA0)				
<i>Address : C005_A038h</i>				
[31:0]	R	REGSTATA[31:0]	Read StatA Register[31:0] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. A enable.	32h0
StatA Register[63:32] (REGSTATA1)				
<i>Address : C005_A03Ch</i>				
[31:0]	R	REGSTATA[63:32]	Read StatA Register[63:32] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. A enable.	32h0
StatA Register[95:64] (REGSTATA2)				
<i>Address : C005_A040h</i>				
[31:0]	R	REGSTATA[95:64]	Read StatA Register[95:64] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. A enable.	32h0
StatA Register[127:96] (REGSTATA3)				
<i>Address : C005_A044h</i>				
[31:0]	R	REGSTATA[127:96]	Read StatA Register[127:96] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. A enable.	32h0
StatA Register[159:128] (REGSTATA4)				
<i>Address : C005_A048h</i>				
[31:0]	R	REGSTATA[159:128]	Read StatA Register[159:128] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. A enable.	32h0
StatA Register[191:160] (REGSTATA5)				
<i>Address : C005_A04Ch</i>				
[31:0]	R	REGSTATA[191:160]	Read StatA Register[191:160] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. A enable.	32h0
StatB Register[31:0] (REGSTATB0)				
<i>Address : C005_A050h</i>				
[31:0]	R	REGSTATB[31:0]	Read StatA Register[31:0] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. B enable.	32h0

Bit	R/W	Symbol	Description	Reset Value
StatB Register[63:32] (REGSTATB1)				
<i>Address : C005_A054h</i>				
[31:0]	R	REGSTATB[63:32]	Read StatA Register[63:32] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. B enable.	32h0
StatB Register[95:64] (REGSTATB2)				
<i>Address : C005_A058h</i>				
[31:0]	R	REGSTATB[95:64]	Read StatA Register[95:64] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. B enable.	32h0
StatB Register[127:96] (REGSTATB3)				
<i>Address : C005_A05Ch</i>				
[31:0]	R	REGSTATB[127:96]	Read StatA Register[127:96] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. B enable.	32h0
StatB Register[159:128] (REGSTATB4)				
<i>Address : C005_A060h</i>				
[31:0]	R	REGSTATB[159:128]	Read StatA Register[159:128] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. B enable.	32h0
StatB Register[191:160] (REGSTATB5)				
<i>Address : C005_A064h</i>				
[31:0]	R	REGSTATB[191:160]	Read StatA Register[191:160] when ENBCAPUSERSTAT is TRUE, start-of-block pulse is HIGH and channel status ch. B enable.	32h0

Section 31. PDM

31.1 Overview

Pulse-density modulation (hereafter, PDM) is a form of modulation used to represent an analog signal with digital data. In a PDM signal, specific amplitude values are not encoded into pulses of different size as they would be in PCM. Instead, it is the relative density of the pulses that corresponds to the analog signal's amplitude. The output of a 1-bit DAC is the same as the PDM encoding of the signal. The PDM in NXP4330D/Q is a block that receives the PDM signals from a exterior digital Microphone (with 1bit IO) and demodulates the 1bit digital signals and returns a original amplitude. The PDM supports DMA interface.

Note : if users want to use this module, contact to Nexell.

31.1.1 Features

- Supports receiving 2 channel audio data with 1 data pin
- 1 output clock pin
- 2 data pins (total 4 channel accepts available)
- Fixed output clock frequency
- Supports select of the timing of data capture
- Supports DMA interface
- Supports a user configuration of Coefficient. (Butterworth Low-pass Filter)

31.1.2 Block Diagram

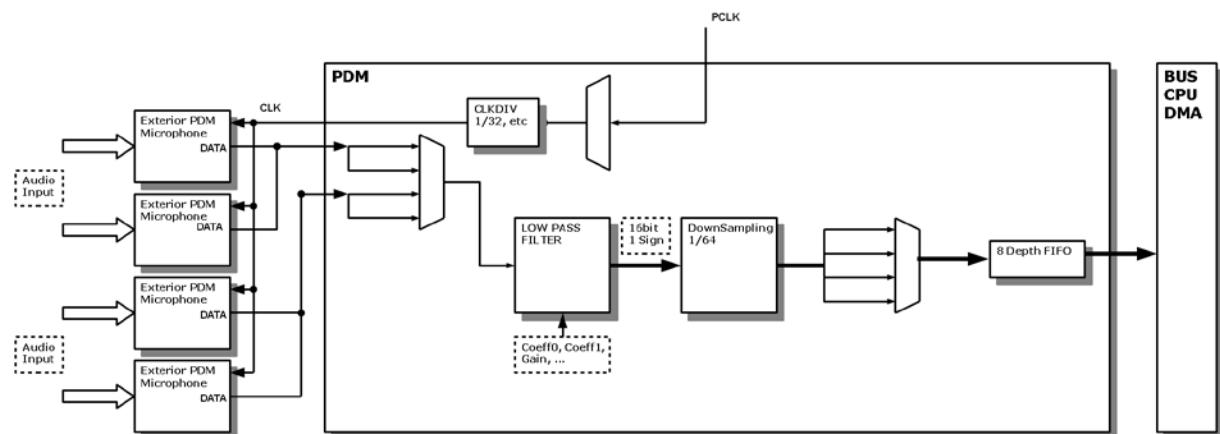


Figure 31-1. PDM Block Diagram

31.1.3 PDM Application Note

31.1.3.1 Butterworth Filter Configuration

The PDM has registers for the Butterworth Filter coefficients. Users can adjust filter coefficients and use own filter for PDM Input data.

Following a example of PDM configuration

- filtertype = Butterworth

- passtype = Lowpass
- ripple = -80 dB
- order = 2
- samplerate = 3072000
- corner1 = 8000
- corner2 =
- adzero =
- logmin =

Register Name	Bit	Symbol	Value
PDM_GAIN0	[31:16]	GAIN X (4)	276 (0x0114)
	[15:0]	GAIN X (2)	138 (0x008a)
PDM_GAIN1	[31:16]	GAIN X (-4)	-276 (0xfeec)
	[15:0]	GAIN X (-2)	-138 (0xffff76)
PDM_COEFF	[31:16]	CPU_COEFF1	16194
	[15:0]	CPU_COEFF0	-8004

PDM Filter Configuration Example

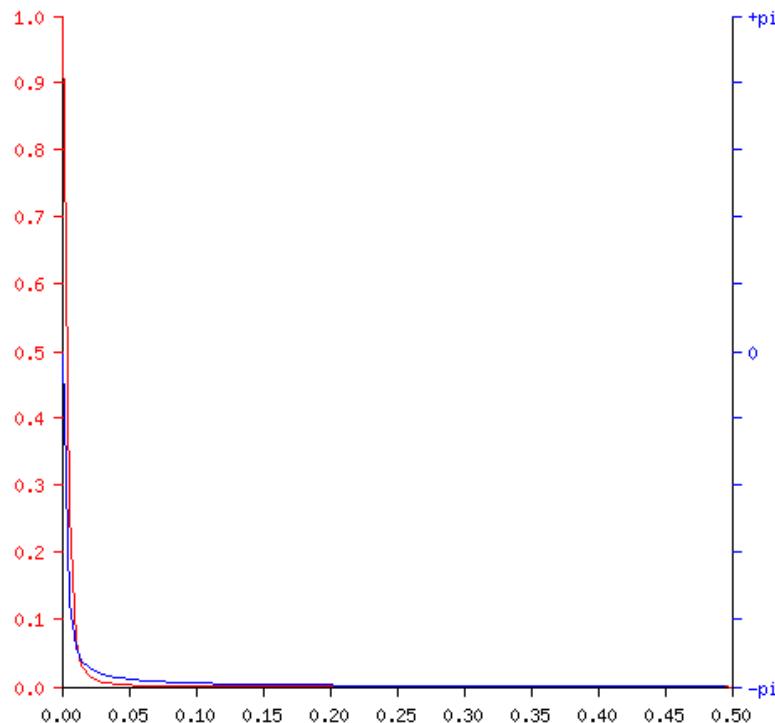


Figure 31-2. Filter Characteristics

31.2 Register Summary

Bit	R/W	Symbol	Description	Reset Value								
PDM Control register (PDM_CTRL) Address :C001 4000h												
[31:23]	R	RESERVED	Reserved	9'b0								
[22:16]	R/W	CPU_OVERSAMPLE	Specifies the value of over sampling.	7'b0								
[15:12]	R	RESERVED	Reserved	4'b0								
[11:8]	R/W	CPU_SEL_SHIFT	Specifies the position of shift strobe. (output clock position)	4'b0								
[7:3]	R	RESERVED	Reserved	5'b0								
[2]	R/W	CPU_DMACHMODE	Enable DMA Mode 0 : Disable 1 : Enable	1'b0								
[1]	R/W	CPU_ENB	Enable PDM 0 : Disable 1 : Enable	1'b0								
[0]	R/W	CPU_INIT	Software Reset	1'b0								
PDM gain 0 register (PDM_gain0) Address :C001 4004h												
[31:16]	R/W	GAIN X (4)	Specifies the value of Filter Gain x 4	16'b0								
[15:0]	R/W	GAIN X (2)	Specifies the value of Filter Gain x 2	16'b0								
PDM gain 1 register (PDM_gain1) Address :C001 4008h												
[31:16]	R/W	GAIN X (-4)	Specifies the value of Filter Gain x (-4)	16'b0								
[15:0]	R/W	GAIN X -(2)	Specifies the value of Filter Gain x (-2)	16'b0								
PDM coefficient register (PDM_coeff) Address :C001 400Ch												
[31:16]	R/W	CPU_COEFF1	Specifies the value of Filter Coefficient 1	16'b0								
[15:0]	R/W	CPU_COEFF0	Specifies the value of Filter Coefficient 0	16'b0								
PDM data register (PDM_data) Address :C001 4010h												
[31:0]	R	PDM_DATA	Demodulated amplitude of input PDM signals Single Mode <table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> </table> Dual Mode <table border="1"> <tr><td>1</td><td>0</td></tr> <tr><td>3</td><td>2</td></tr> </table>	1	0	1	0	1	0	3	2	
1	0											
1	0											
1	0											
3	2											
PDM control register 1 (PDM_ctrl1) Address :C001 4014h												
[31:19]	R	RESERVED	Reserved	13'b0								
[18:16]	R/W	CPU_NUM_SHIFT_CLOCK	Specifies the number of output clock shift	3'b0								
[15:8]	R/W	CPU_NUM_CLOCK	Specifies the toggle position of output clock	8'b0								

Bit	R/W	Symbol	Description	Reset Value
[7:0]	R/W	CPU_SAMPLE_POS	Specifies the sampling position for PDM Input data	8'b0
PDM interrupt control register (PDM_irqctrl)				
Address :C001 4018h				
[31:7]	R	RESERVED	Reserved	25'b0
[6]	R	INTPEND	Interrupt Pending Bit	
[5]	W	INTPEND_CLR	Write 1 : Interrupt Pending Clear	
[4:0]	R/W	IRQ_COUNT	Specifies the count for PDM Interrupt. The interrupt is occurred by internal FIFO write counter. User must use this count only for PIO mode. 0 : Interrupt Disable	5'b0

Section 32. Display Architecture

32.1 Overview

32.1.1 Features

- 2 Multi-layer Controller, 2 Display Controller
- 1 HDMI, 1 LVDS, 1 MIPI DSI
- Supports TFT/MPU LCD Interface, HDMI, LVDS, MIPI DSI output formats
- Supports that transmits the same image through different outputs at the same time .

32.1.2 Block Diagram

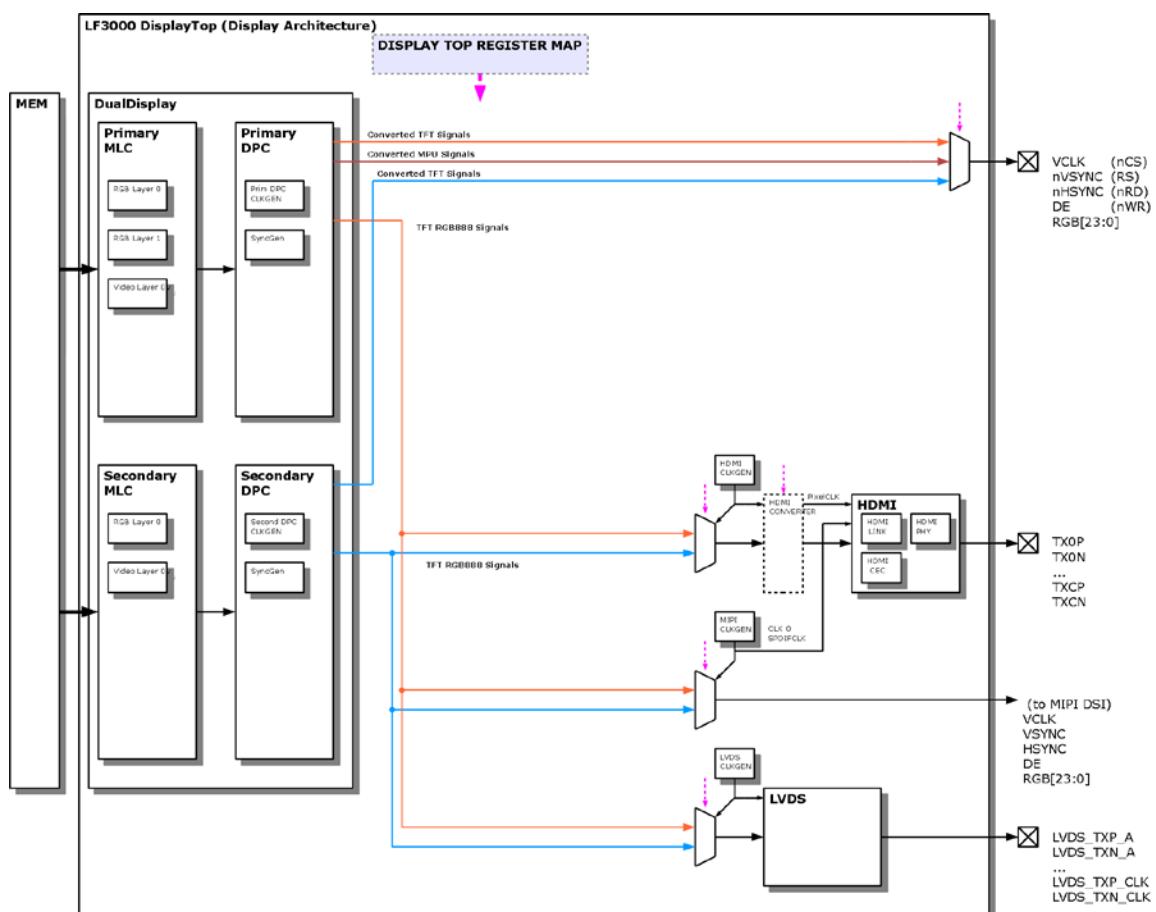


Figure 32-1. NXP4330D/Q Display Architecture Block Diagram

32.2 TFT/MPU Interface

Things can be output through the LCD Interface as follows.

- 1) Primary DPC (TFT Interface)
- 2) Primary DPC (i80 MPU Interface)
- 3) Secondary DPC (TFT Interface)

Note : There is no i80 MPU Interface in Secondary DPC

Users can select one of them by setting the *TFT_MUXCTRL* register.

32.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
DISPLAYTOP HDMI MUX Control register (HDMI_MUXCTRL)				
<i>Address :C010 1004h</i>				
[31]	R/W	HDMI_MUXENB	MUX Enable 0: MUX Disable 1 : MUX Enable	1'b0
[30:2]	R/W	RESERVED	Reserved	29'b0
[1:0]	R/W	HDMI_MUXSEL	MUX Select 0: Primary DPC 1 : Secondary DPC 2-3 : Reserved	2'b0
DISPLAYTOP LVDS MUX Control register (LVDS_MUXCTRL)				
<i>Address :C010 100Ch</i>				
[31]	R/W	LVDS_MUXENB	MUX Enable 0: MUX Disable 1 : MUX Enable	1'b0
[30:2]	R/W	RESERVED	Reserved	29'b0
[1:0]	R/W	LVDS_MUXSEL	MUX Select 0: Primary DPC 1 : Secondary DPC 2-3 : Reserved	2'b0
DISPLAYTOP HDMI sync Control register 0 (HDMI_SYNCCTRL0)				
<i>Address :C010 1014h</i>				
[31]	R/W	HDMI_VCLK_SEL	Must set this value to 0 0: HDMI PHY's pixel clock used for HDMI Operation 1 : Never set this value	1'b0
[30:16]	R/W	RESERVED	Reserved	15'b0
[15:0]	R/W	HDMI_VSYNCSTART	Specifies the start line of i_v_sync for the HDMI Link	16'b0
DISPLAYTOP HDMI sync Control register 1 (HDMI_SYNCCTRL1)				
<i>Address :C010 1018h</i>				
[31:16]	R/W	RESERVED	Reserved	16'b0
[15:0]	R/W	HDMI_HACTIVESTART	Specifies the start position(h_line) of h_active for the HDMI Link	16'b0
DISPLAYTOP HDMI sync Control register 2 (HDMI_SYNCCTRL2)				
<i>Address :C010 101Ch</i>				
[31:16]	R/W	RESERVED	Reserved	16'b0
[15:0]	R/W	HDMI_HACTIVEEND	Specifies the end position of h_active for the HDMI Link	16'b0
DISPLAYTOP HDMI sync Control register 3 (HDMI_SYNCCTRL 3)				
<i>Address :C010 1020h</i>				
[31:16]	R/W	HDMI_VSYNCHSEND	Specifies the end position of i_v_sync for the HDMI Link	16'b0
[15:0]	R/W	HDMI_VSYNCHSSTART	Specifies the start position of i_v_sync for the HDMI Link	16'b0
DISPLAYTOP TFT MUX Control register (TFT_MUXCTRL)				
<i>Address :C010 1024h</i>				
[31:2]	R/W	RESERVED	Reserved	30'b0

Bit	R/W	Symbol	Description	Reset Value
[1:0]	R/W	TFT_MUXSEL	MUX Select 0: Primary DPC (TFT) 1 : Priamry DPC (i80 MPU) 2: Secondary DPC (TFT) 3 : Reserved (Never use this value)	16'b0

Section 33. Multi Layer Controller (MLC)

33.1 Overview

The user screen is composed of complex components - RGB pictures, moving pictures, etc. These individual components have unique formats and are stored in their own memory spaces. The Multi Layer Controller (hereinafter, MLC) of NXP4330D/Q reads and compounds various screen components in terms of Hardware, to organize a desired screen and transmits the result to the Display controller.



Figure 33-1. Concept of Multi Layer Controller

33.1.1 Features

- Dual register-set architecture
- Two/One RGB layers and one Video layer
(Primary Display : 2 RGB layer, Secondary Display : 1 RGB layer)
- RGB layers can be used as 3D layers.
- Various pixel formats
 - RGB layer: RGB/BGR 332, 444, 555, 565, 888 with/without Alpha
 - Video layer: 2D Separated YUV 4:4:4, 4:2:2, 4:2:0, Linear YUV 4:2:2(YUYV)
- Various blending effects between layers
 - Per-layer or Per-pixel Alpha blending, Transparency, Inverse color
- Free layer position and size in pixel units
- Hardware clipping
- Vertical flip
- Video layer priority

- Gamma Correction
- Configurable Burst Length (*LOCKSIZE*, RGB layer)
- Scale-up/down (Video layer only)
 - Bilinear interpolation, Nearest neighbor sampling scale-up/down
- Color control (Video layer only)
 - Brightness, Contrast, Hue, Saturation

33.1.2 Block Diagram

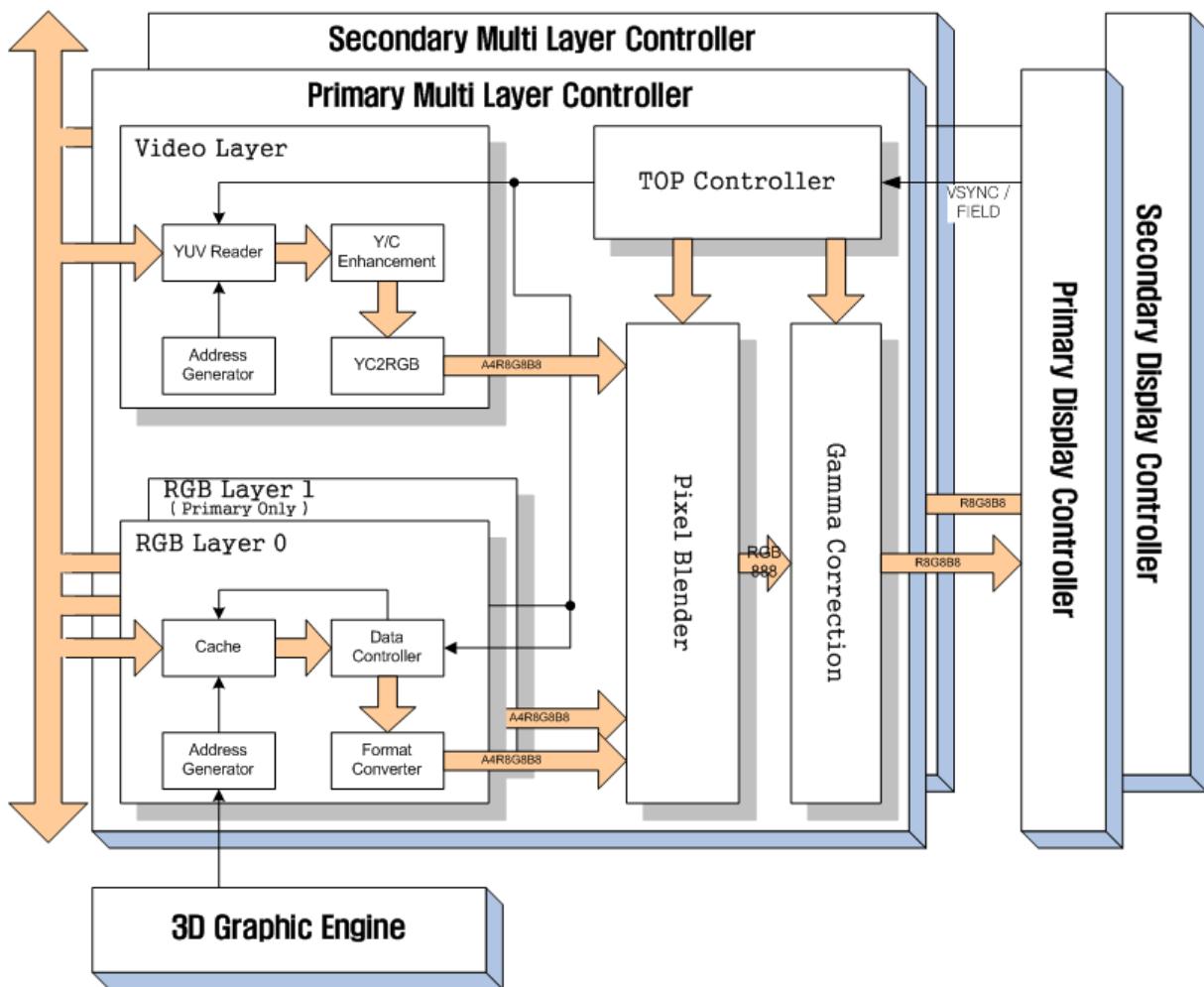


Figure 33-2. MLC Block Diagram

The MLC consists of three RGB layers and one Video layer. In the MLC, positions, pixel formats and various effects can be configured according to each layer. The Video layer supports the Scale function to display video images on various screen areas at certain sizes and various color control functions to provide optimal images.

33.2 Dual Register Set Architecture

The MLC of NXP4330D/Q has dual register set architecture with a current working register group and a user side register group. All users can set registers via the user side register group. If the dirty flag is set as '1' after the user writes a desired setting to a register in the user side register group, the MLC copies the user side register group to the current working register group at the point when vertical sync occurs. Then the dirty flag is cleared to '0' and the user can continue to progress the next setting. In this way the changes in all registers are synchronized and applied to vertical sync so that the user can hide any abnormal screen, even if the user changes a register at any point.

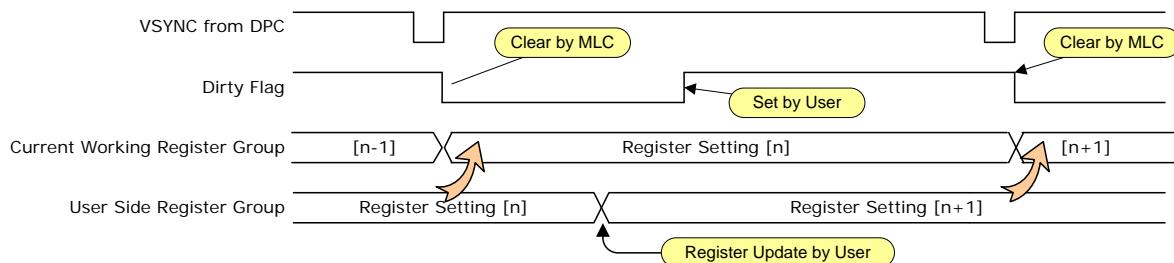


Figure 33-3. Dual Register Set Architecture

The Top controller and three layers of the MLC have separate dirty flag bits. Each dirty flag reflects the changes of the registers pertaining to the corresponding group.

Dirty flag	Numbers	Registers to be Affected
Top controller	1	MLCCONTROLT, MLCSCREENSIZE, MLCBGCOLOR
RGB layer	2	MLCLEFTRIGHTn, MLCTOPBOTTOMMn, MLCCONTROLn, MLCHSTRIDEn, MLCVSTRIDEn, MLCTPCOLORn, MLCINVCOLORn, MLCAADDRESSn, MLCLEFTRIGHTn_0, MLCTOPBOTTOMn_0, MLCLEFTRIGHTn_1, MLCTOPBOTTOMn_1
Video layer	1	MLCLEFTRIGHT2, MLCTOPBOTTOM2, MLCCONTROL2, MLCVSTRIDE2, MLCTPCOLOR2, MLCAADDRESS2, MLCAADDRESSCB, MLCAADDRESSCR, MLCVSTRIDECB, MLCVSTRIDECR, MLHSCALE, MLCVSCALE, MLCLUENH, MLCCHENH0, MLCCHENH1, MLCCHENH2, MLCCHENH3

Table 33-1. Dirty flag

33.3 MLC Global parameters

This section describes how to set the global parameters of the MLC.

Function	Symbol	Bit width	Register	Brief description
Priority	PRIORITY	2	MLCCONTROLT[9:8]	Specifies the priority of the Video layer.
Dual register set	DIRTYFLAGT	1	MLCCONTROLT[3]	Dirty Flag for MLC top controller.
Enable	MLCENB	1	MLCCONTROLT[1]	Specifies whether or not to enable MLC
Scan mode	FIEEDENB	1	MLCCONTROLT[0]	Specifies whether or not to enable Interlace mode
Screen size	SCREENWIDTH	12	MLCSCREENSIZE[11:0]	Specifies 'the whole screen width - 1'.
	SCREENHEIGHT	12	MLCSCREENSIZE[27:16]	Specifies 'the whole screen height - 1'.
Background color	BGCOLOR	24	MLCBGCOLOR[23:0]	Specifies the background color to be displayed on the screen in areas not covered by any of the layers
RGB Gamma	RGBGAMMAENB	1	MLCGAMMACONT[1]	Gamma Enable for the RGB region
Dithering	DITHERENB	1	MLCGAMMACONT[0]	Dithering Enable for the result of Gamma correction
Video Gamma	VIDEOGAMMAENB	1	MLCGAMMACONT[4]	Gamma Enable for the Video region
Alpha select	ALPAHASELECT	1	MLCGAMMACONT[5]	Allocate the Alpha blended region of RGB layer and Video layer to the Video region or the RGB region

Table 33-2. Top controller registers

33.3.1 Screen size

The Screen size function enables users to specify the width and height of the whole screen to be displayed. The values of 'the whole screen width – 1' and 'the whole height – 1' are set to the **SCREENWIDTH** and **SCREENHEIGHT**, respectively. Since each of the **SCREENWIDTH** and **SCREENHEIGHT** has 12-bit size units, the maximum available resolution is 2048 x 2048 pixels. The screen size is determined by setting the size in frame units regardless of whether the display is progressive or interlace.

33.3.2 Priority

The MLC consists of three RGB layers and one Video layer. Among the RGB layers, Layer 0 has the highest priority and Layer 1 has the lowest priority. These priorities cannot be changed, but the priority of the Video layer can be adjusted via the **PRIORITY** parameter at the user's discretion.

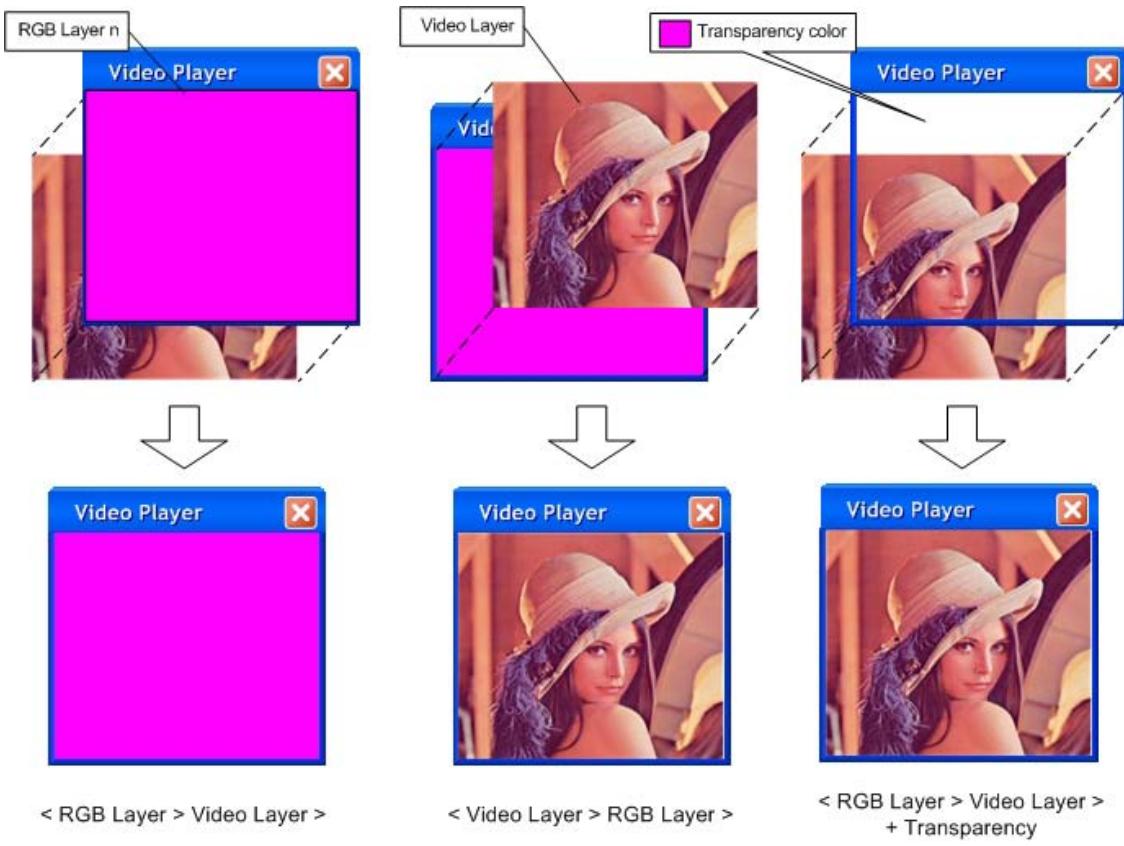


Figure 33-4. Layer priority

33.3.3 Field Mode

The MLC of NXP4330D/Q supports an interlace display as well as a progressive display. All registers of the MLC should be set in frame units. For the progressive display, the **FIELDENB** bit of the Top controller should be set as '0'. For the interlace display, the **FIELDENB** bit of the Top controller should be set as '1'. For example, a 720 x 480 progressive display and a 720 x 480 interlace display have the same settings, except for the setting of the **FIELDENB** bit.

33.3.4 Background color

Each layer of the MLC can be positioned at any place on the screen. Therefore, it is possible for there to be an area not contained in any of the layers actually on the screen. The default color displayed in this area is called the background color and the background color is set to **BGCOLOR**. If all layers are disabled, only the background color is displayed on the screen.

The bpp of the **BGCOLOR** is 24 and has the following format:

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Background color	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

Table 33-3. Background color format

33.4 Per-layer parameters

Function	Symbol	Bit width	Register	Brief description
Enable	LAYERENB	1	MLCCONTROLn[5]	Specifies whether or not to enable this layer.
Dual register set	DIRTYFLAG	1	MLCCONTROLn[4]	Dirty flag for this layer
Lock control	LOCKSIZE ¹⁾	2	MLCCONTROLn[13:12]	Specifies lock size for memory access.
Position	LEFT	12	MLCLEFTRIGHTn[27:16]	Specifies x-coordinate of upper-left corner.
	TOP	12	MLCTOPBOTTOMn [27:16]	Specifies y-coordinate of upper-left corner.
	RIGHT	12	MLCLEFTRIGHTn [11:0]	Specifies x-coordinate of lower-right corner.
	BOTTOM	12	MLCTOPBOTTOMn[11:0]	Specifies y-coordinate of lower-right corner.
InValid Position0	INVLDENB ¹⁾	1	MLCLEFTRIGHTn_0[28]	InValid0 Area Enable
	LEFT ¹⁾	12	MLCLEFTRIGHTn_0[26:16]	Specifies x-coordinate of upper-left corner.
	TOP ¹⁾	12	MLCTOPBOTTOMn_0[26:16]	Specifies y-coordinate of upper-left corner.
	RIGHT ¹⁾	12	MLCLEFTRIGHTn_0[10:0]	Specifies x-coordinate of lower-right corner.
	BOTTOM ¹⁾	12	MLCTOPBOTTOMn_0[10:0]	Specifies y-coordinate of lower-right corner.
InValid Position1	INVLDENB ¹⁾	1	MLCLEFTRIGHTn_1[28]	InValid1 Area Enable
	LEFT ¹⁾	12	MLCLEFTRIGHTn_1[26:16]	Specifies x-coordinate of upper-left corner.
	TOP ¹⁾	12	MLCTOPBOTTOMn_1[26:16]	Specifies y-coordinate of upper-left corner.
	RIGHT ¹⁾	12	MLCLEFTRIGHTn_1[10:0]	Specifies x-coordinate of lower-right corner.
	BOTTOM ¹⁾	12	MLCTOPBOTTOMn_1[10:0]	Specifies y-coordinate of lower-right corner.
Alpha blending	BLENDENB	1	MLCCONTROLn[2]	Specifies whether or not to enable alpha blending.
	ALPHA	8	MLCTPCOLORn[31:24]	Specifies alpha blending factor.
Color inversion	INVENB ¹⁾	1	MLCCONTROLn[1]	Specifies whether or not to enable color inversion.
	INVCOLOR ¹⁾	24	MLCINVCOLOR[23:0]	Specifies the color to be used for color inversion.
Transparency	TPENB ¹⁾	1	MLCCONTROLn[0]	Specifies whether or not to enable transparency.
	TPCOLOR ¹⁾	24	MLCTPCOLORn[23:0]	Specifies the color to be used as transparency color.
Address generation	ADDRESS	32	MLCADDRESSn[31:0]	Specifies the base address of image buffer.
	HSTRIDE ¹⁾	32	MLCHSTRIDEn[31:0]	Specifies the horizontal stride in bytes.
	VSTRIDE	32	MLCVSTRIDEn[31:0]	Specifies the vertical stride in bytes.
	ADDRESSCB ²⁾	32	MLCADDRESSCB[31:0]	Specifies the base address of Cb image buffer
	ADDRESSCR ²⁾	32	MLCADDRESSCR[31:0]	Specifies the base address of Cr image buffer.
	VSTRIDECB ²⁾	32	MLCVSTRIDECB[31:0]	Specifies the vertical stride in bytes for Cb image buffer.
	VSTRIDECR ²⁾	32	MLCVSTRIDECR[31:0]	Specifies the vertical stride in bytes for Cr image buffer.
Note 1) Only exists in RGB layers				
Note 2) Only exists in Video layer				

Table 33-4. RGB layer registers

33.4.1 Enable

Each layer has a LAYERENB bit to enable/disable each layer of NXP4330D/Q at a certain point. If the *LAYERENB* bit is set as '1', the relevant layer becomes on. If the *LAYERENB* bit is set as '0', the relevant layer becomes off and the other layer setting registers are not used. Since the setting of the *LAYERENB* bit is reflected by a dirty flag, the layer is toggled on/off in accordance with a VSYNC signal, even if the user controls the *LAYERENB* bit at a certain point.

33.4.2 Lock control

Each RGB layer can adjust the data size to be read at any one time when a memory read is requested through the Lock control. The *LOCKSIZE* can specify 4, 8 or 16 and the unit size is 16 bytes. Therefore, if the *LOCKSIZE* is 4, the data size to be read at any one time is 64 bytes. For a resolution of 1280 x 1024 or higher, it is recommended to set the *LOCKSIZE* as 16.

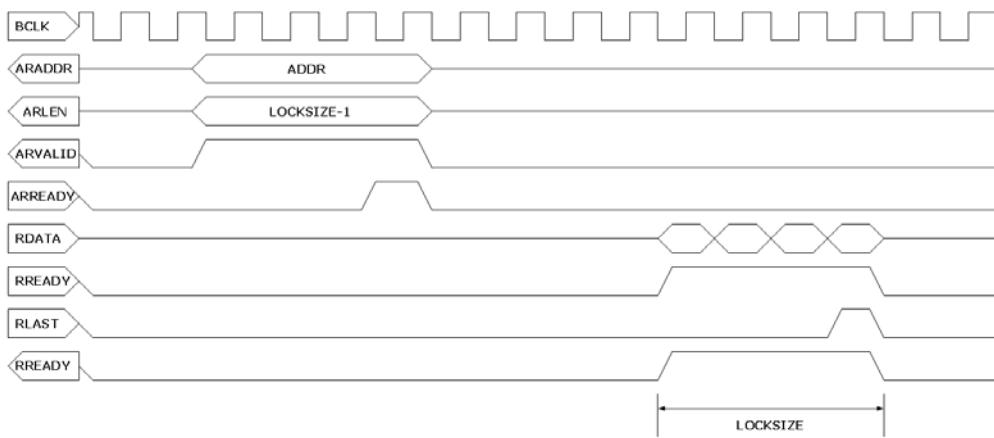


Figure 33-5. Lock timing

33.4.3 Position

The Position function enables users to specify the top-left (*LEFT* and *TOP*) and the bottom-right (*RIGHT* and *BOTTOM*) coordinates. Each coordinate can be positioned at any point within the range from -2048 to 2047, but only the layer contained in the area from (0, 0) and (ScreenWidth – 1, ScreenHeight – 1) is displayed on the actual screen. The MLC of NXP3200 supports H/W clipping for any area outside of the screen area, so users do not need to carry out additional clipping processing. In addition, the MLC does not read the data in the clipped area and the area hidden by upper layer from memory for effective use of the memory bandwidth.

RGB Layer could appoint three invisible areas without using certain color. Appointed area is not read from Memory.

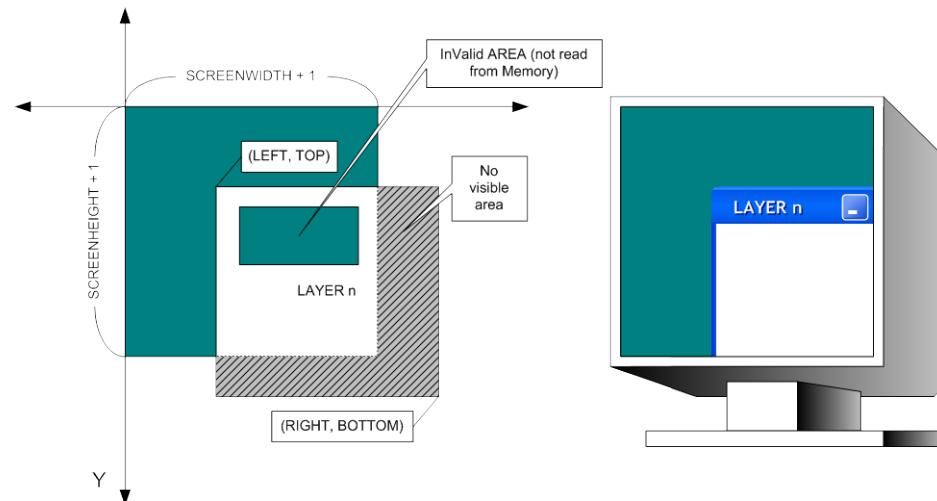


Figure 33-6. Layer position

33.4.4 Pixel format

33.4.4.1 RGB layer format

Each RGB layer supports various formats and the formats are listed in Table 33-5.

R : Red, G : Green, B : Blue, A : Alpha, X : Not used

Pixel format	FORMAT[15:0]	Bpp	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
R5G6B5	4432h	16			R1[4:0]				G1[5:0]				B1[4:0]				R0[4:0]				G0[5:0]				B0[4:0]																	
B5G6R5	C432h	16				B1[4:0]				G1[5:0]				R1[4:0]				B0[4:0]				G0[5:0]				R0[4:0]																
X1R5G5B5	4342h	16					R1[4:0]				G1[4:0]			B1[4:0]					R0[4:0]				G0[4:0]				B0[4:0]															
X1B5G5R5	C342h	16					B1[4:0]				G1[4:0]			R1[4:0]					B0[4:0]				G0[4:0]				R0[4:0]															
X4R4G4B4	4211h	16						R1[3:0]			G1[3:0]			B1[3:0]						R0[3:0]			G0[3:0]			B0[3:0]																
X4B4G4R4	C211h	16							B1[3:0]		G1[3:0]			R1[3:0]						B0[3:0]			G0[3:0]			R0[3:0]																
X8R3G3B2	4120h	16								R1[2:0]		G1[2:0]	B1[1:0]								R0[2:0]	G0[2:0]	B0[1:0]																			
X8B3G3R2	C120h	16								B1[2:0]		G1[2:0]	R1[1:0]								B0[2:0]	G0[2:0]	R0[1:0]																			
A1R5G5B5	3342h	16	A1			R1[4:0]			G1[4:0]			B1[4:0]		A0		R0[4:0]			G0[4:0]			B0[4:0]																				
A1B5G5R5	B342h	16	A1			B1[4:0]			G1[4:0]			R1[4:0]		A0		B0[4:0]			G0[4:0]			B0[4:0]																				
A4R4G4B4	2211h	16		A1[3:0]		R1[3:0]			G1[3:0]		B1[3:0]			A0[3:0]		R0[3:0]			G0[3:0]			B0[3:0]																				
A4B4G4R4	A211h	16		A1[3:0]		B1[3:0]			G1[3:0]		R1[3:0]			A0[3:0]		B0[3:0]			G0[3:0]			R0[3:0]																				
A8R3G3B2	1120h	16			A1[7:0]				R1[2:0]		G1[2:0]	B1[1:0]			A0[7:0]				R0[2:0]	G0[2:0]	B0[1:0]																					
A8B3G3R2	9120h	16			A1[7:0]				B1[2:0]		G1[2:0]	R1[1:0]			A0[7:0]				B0[2:0]	G0[2:0]	R0[1:0]																					
R8G8B8	4653h ¹⁾	24				B1[7:0]				R0[7:0]						G0[7:0]						B0[7:0]																				
B8G8R8	C653h ¹⁾	24				R1[7:0]				B0[7:0]						G0[7:0]						R0[7:0]																				
X8R8G8B8	4653h ¹⁾	32								R[7:0]						G[7:0]						B[7:0]																				
X8B8G8R8	C653h ¹⁾	32								B[7:0]						G[7:0]						R[7:0]																				

Pixel format	FORMAT[15:0]	Bpp	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A8R8G8B8	0653h	32	A[7:0]				R[7:0]				G[7:0]				B[7:0]																			
A8B8G8R8	8653h	32	A[7:0]				B[7:0]				G[7:0]				R[7:0]																			

Note 1) The format settings for R8G8B8 & X8R8G8B8 and B8G8R8 & X8B8G8R8 are the same. However, the **HSTRIDES** for R8G8B8 and B8G8R8 should be set as '3' because they are in 24 bpp modes, while the **HSTRIDES** for X8R8G8B8 and X8B8G8R8 should be set as '4'.

Table 33-5. RGB layer format

The above formats are converted into A8R8G8B8 and are managed in each RGB layer. Each color component is converted into 8-bit size. The color components with the size smaller than 8-bit are extended to 8-bit size from the highest bit repeatedly. For example, the color with 5-bit size is converted to {[4:0], [4:2]} and the color with 3-bit size is converted to {[2:0], [2:0], [2:1]}. Each layer compares the color component internally extended with **TPCOLOR** or **INVCOLOR**. Therefore, a user should set the color that each color format is extended in R8G8B8 in **TPCOLOR** and **INVCOLOR**.

33.4.4.2 Video layer format

The Video layer manages YUV data and supports the linear YUV format and the 2D block addressing separated YUV format.

FORMAT[1:0]	Pack mode	Type	Y:UV	Addressing mode
0	Separate Y/U/V	YUV	4:2:0	2D Block
1	Separate Y/U/V	YUV	4:2:2	2D Block
2	Non-separate	YUV	4:2:2	Linear
3	Separate Y/U/V	YUV	4:4:4	2D Block
4	Separate Y/U/V	YUV	4:2:2	2D Block
5	Separate Y/U/V	YUV	4:2:0	2D Block

Table 33-6. Video layer format

The MLC of **NXP4330D/Q** uses the A4R8G8B8 format internally. Therefore, the Video layer also reads YUV data from the memory and converts into RGB data internally. The formulas with which the Video layer converts YUV data into RGB data are as follows:

- ※ The formula for YCbCr to RGB conversion
 - $R = Y + (1.4020 * Cr)$
 - $G = Y - (0.34414 * Cb) + (0.71414 * Cr)$
 - $B = Y + (1.7720 * Cb)$

Linear YUV 422 format

The video layer supports the YUYV format as a linear YUV format whose Y data (luminance data) exists at every pixel, but Cb and Cr (chrominance) data exist separately over two pixels. Therefore two pixels share the same Cb/Cb data. Hence the Video layer has 2-pixel data per 32-bit and manages it in 2-pixel units.

Pixel format	FORMAT[2:0]	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YUYV	2	Cr[7:0]				Y1[7:0]				Cb[7:0]				Y0[7:0]																			

Table 33-7. YUYV format

2D block addressing separated Y/U/V format

In the 2D block addressing separated Y/U/V format, each of Y, U and V exists at separate memory spaces. In addition, the format is divided into 444, 422 and 420 in proportion to U and V for Y. The 2D block addressing separated YUV format is the 2D block addressing format, and each component has a size of 64 x 32 and linearity in block units. These features provide NXP4330D/Q's unique memory format, to enhance the effectiveness of memory access when NXP4330D/Q manages data in macro block units through an algorithm to compress/decompress images like MPEG files.

According to each format, Y, U and V correspond to 2 x 2 pixels as follows:

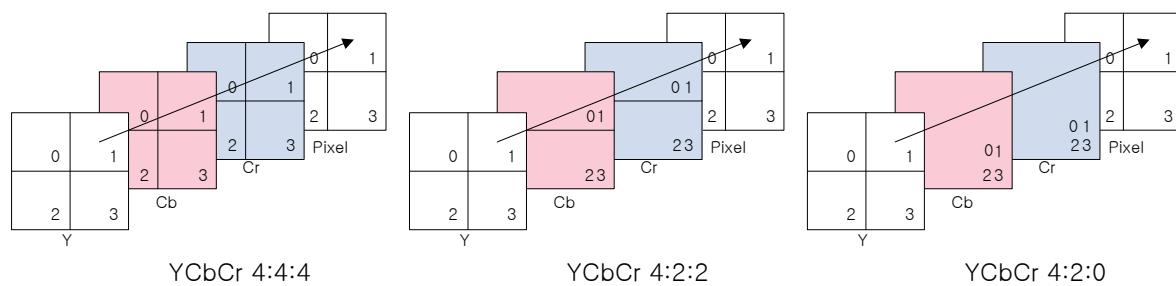
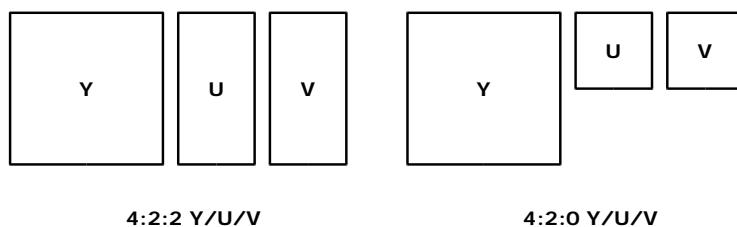


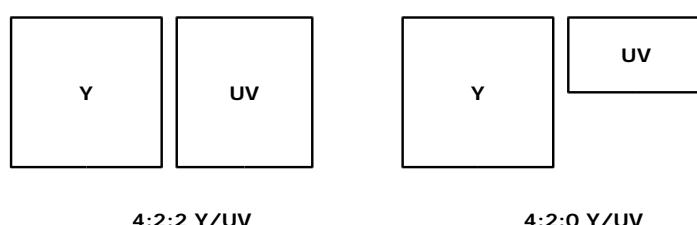
Figure 33-7. Separated YUV format

In the memory, each of Y, U and V exists at separated memory space.



2D block addressing separated Y/UV format

In the 2D block addressing separated Y/UV format, Y, UV exists at separate memory spaces. In addition, the format is divided into 422 and 420. In the UV block, data exists linearly.



Pixel format	FORMAT[2:0]	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YUYV	4 or 5																																

33.4.4.3 Layer blending

MLC consists of three RGB layers and one Video layer. Each RGB layer supports Transparency, Color Inversion and Alpha Blending functions but the Video layer only supports Alpha Blending functions. The Color Inversion and the Alpha

Blending functions are only applied to between layers and not to background.

The Transparency function enables users to specify a particular color and handle the color as a transparent color. Therefore, an area filled with a transparent color shows through the lower layer and shows the layer as it is. Like the Transparency function, the Color Inversion function shows through the lower layer and shows the layer as it is, but the function is different in that it inverts and projects the layer's color. The Transparency and Color Inversion functions are useful for the implementation of the Cursor layer as shown in the figure below:

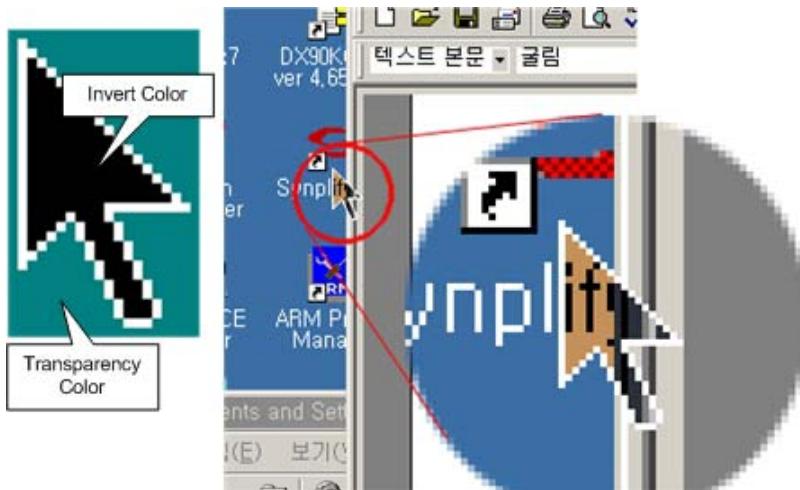


Figure 33-8. Transparency & Color inversion

The Transparency and the Color Inversion effects are applied by setting each of the **TPENB** and the **INVENB** bits of an RGB Layer as '1'. Both **TPCOLOR** and **INVCOLOR** have R8G8B8 formats.

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPCOLOR / INVCOLOR	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

Table 33-8. TPCOLOR & INVCOLOR format

The Alpha Blending function enables users to adjust the transparency of a desired layer. The Alpha level is adjusted by **ALPHA[7:0]** and can be specified in the range between 0 and 255. If the Alpha level is 255, it means it is fully opaque. If the level is 0, it indicates full transparency. In addition, each RGB layer can use a pixel format including Alpha and is applied in Alpha per pixel units. In the Per-layer alpha or the Per-pixel Alpha formats, the Alpha Blending effect can only be applied by setting the **BLENDENB** bit as '1'.

Only one layer can use the alpha blending function at the same time, and when the alpha blending function of one layer is enabled, the alpha blending function of the other layers must be disabled.

All layers use the A8R8G8B8 format internally and the formula for Alpha Blending is as follows:

※ The formula for Alpha blending function

If alpha is 0 then α is 0, else α is alpha + 1

Result color = this layer color * α / 256 + lower layer color * (256 - α) / 256

In the A4 format, α = alpha*16 + alpha (0 \leq alpha $<$ 16)



Figure 33-9. Alpha blending

33.4.5 Address generation

33.4.5.1 RGB layer address generation

The Address generation function enables users to specify the address and stride of the memory where images are stored.

Stride is divided into horizontal stride and vertical stride. The stride is the unit for increasing an address. The horizontal stride (**HSTRIDE**) is the value to be added to an address whenever its x-coordinate increases, while the vertical stride (**VSTRIDE**) is the value to be added to the address whenever its y-coordinate increases. In general, the horizontal stride is the number of the bytes per pixel and the vertical stride is the value that is the number of bytes per pixel multiplied by an image width. The vertical stride has 2's complement format. Thus a vertical flip function can be implemented by specifying the vertical stride as negative number.

The Image address (**ADDRESS**) usually specifies an address on the top left corner of the image. If the vertical stride for the vertical flip function has a negative number, the **ADDRESS** should specify an address on the bottom left corner of the image.

Vertical flip	Vertical stride	Base address
Off	>0	Address on the top left corner of the image
On	<0	Address on the bottom left corner of the image

Table 33-9. RGB layer address & flip

33.4.5.2 Video layer address generation

In the Video layer, the horizontal stride is not separately specified and is fixed internally. In addition, the vertical stride should always be a positive number. Since the vertical stride is always positive number, the vertical flip function is not supported.

■ Linear YUV 422 format (YUYV)

In the linear YUV 422 format, an address can only be specified by setting its base address and vertical stride. The Base address (**ADDRESS**) specifies the base address in the YUYV image buffer and the vertical stride (**VSTRIDE**) specifies the increment of the address in proportion to the increase of the y-coordinate. The vertical stride should be a positive number.

■ 2D block addressing separated YUV format

In the separated format, each Y, U and V is stored in different addresses of the memory. Thus the address for the Y component is specified by the **ADDRESS3** and the **VSTRIDE3** registers and the address for the U (Cb) and V (Cr)

components is separately specified by the **ADDRESSCB** & the **VSTRIDECB** registers and the **ADDRESSCR** & the **VSTRIDECR** registers. Since **NXP3200** uses segments with 4096 x 4096 pixel sizes in the 2D block addressing format, the vertical stride should be specified as 4096. In addition, the **ADDRESS3/CB/CR** registers set the separate format for segment addresses and not for normal linear addresses. The segment addressing format is listed in Table 33-10.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Address	0	0	1	Index of Segment																							Y coordinate in segment [11:0]		X coordinate in segment [11:0]			

Table 33-10. Segment addressing format

The display array area on the memory map should be specified as a setting Address [31:29] of 4'b001. In addition, the Memory controller should be enabled to use the display array area. See section '5.3.2. Display Array Area' for more detail.

33.4.6 Video layer specific parameters

The Video layer provides the Scale and Color Control functions additionally.

Function	Symbol	Bit width	Register	Brief description
Scale	VFILTERENB	1	MLCVSCALE[28]	Specifies whether or not to vertical scale enable bilinear filter. (This is only applicable to the Y component)
	VFILTERENB_C	1	MLCVSCALE[29]	Specifies whether or not to vertical scale enable bilinear filter. (This is only applicable to the Cb, Cr components)
	HFILTERENB	1	MLCHSCALE[28]	Specifies whether or not to horizontal scale enable bilinear filter. (This is only applicable to the Y component)
	HFILTERENB_C	1	MLCHSCALE[29]	Specifies whether or not to horizontal scale enable bilinear filter. (This is only applicable to the Cb, Cr components)
	HSCALE	23	MLCHSCALE[22:0]	Specifies the horizontal scale ratio.
	VSCALE	23	MLCVSCALE[22:0]	Specifies the vertical scale ratio.
Color control	BRIGHTNESS	8	MLCLUENH[15:8]	Specifies the brightness value.
	CONTRAST	3	MLCLUENH[2:0]	Specifies the contrast value.
	HUECBnA	8	MLCCHENHn[7:0]	Specifies the cosine value for Cb component.
	HUECBnB	8	MLCCHENHn[15:8]	Specifies the sine value for Cb component.
	HUECRnA	8	MLCCHENHn[23:16]	Specifies the sine value for Cr component.
	HUECRnB	8	MLCCHENHn[31:24]	Specifies the cosine value for Cr component.

Table 33-11. Video layer specific parameters

33.4.7 Scale function

The scale-up and the scale-down of the Video layer are determined by the **HSCALE** and **VSCALE** parameters. Each parameter finds and sets the ratio between an input image size and an output image size. The setting formulae for the **HSCALE** and **VSCALE** parameters are as follows:

* The formula for HSCALE and VSCALE

In case of the enlargement by using a Bilinear filter:

- $HSCALE = (\text{source width}-1) * (1<<11) / (\text{destination width}-1)$
- $VSCALE = (\text{source height}-1) * (1<<11) / (\text{destination height}-1)$

```
,else
    • HSCALE = source width * (1<<11) / destination width
        • VSCALE = source height * (1<<11) / destination height
```

Video Layer supports bilinear filtered scaling up and down. Nearest neighbor scaling is also supported. Filter enable signals are separately allocated for Y and C(Cb, Cr) components in case of scaling up/down (H/VFILTERENB, H/VFILTERENB_C)

When bilinear scaling up and down is used, **H/VFILTERENB** and **H/VFILTERENB_C** should be set to '1' for bilinear filtered scaling. When nearest neighbor scaling down is used, **H/VFILTERENB** and **H/VFILTERENB_C** should be set to '0'. The difference in images produced by the Bilinear filter method is as shown in Figure 33-10.



Figure 33-10. Bilinear Filter

33.4.8 Color control

The Video layer supports the Color Control function for output images. A user can adjust Brightness, Contrast, Hue and Saturation to compensate video data colors.

33.4.8.1 Luminance Enhancement

The Video layer can compensate luminance data by adjusting Brightness and Contrast.

The Brightness consists of 256 levels and is set by the **BRIGHTNESS** parameter. The **BRIGHTNESS** parameter has a 2's complement value and can be set between the -128 level and the +127 level. Figure 33-11 shows the image change depending on the Brightness change.

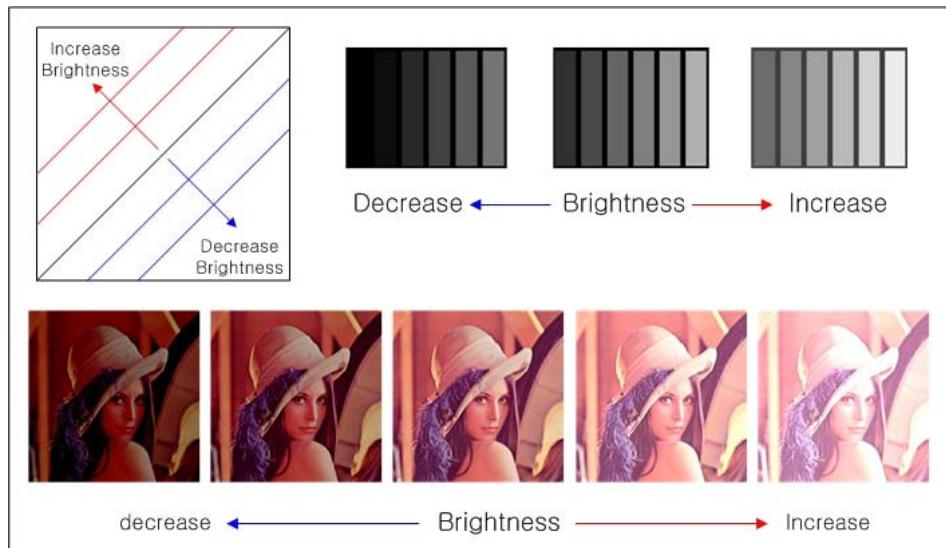


Figure 33-11. Brightness

The Contrast consists of 8 levels and is set by the **CONTRAST** parameter. The Video layer can adjust the contrast from 1.0 to 1.875 in increments of 0.125, but cannot reduce the contrast of the original image. Table 33-12 lists the contrast values corresponding to the **CONTRAST** parameters.

CONTRAST	0	1	2	3	4	5	6	7
Contrast value	1.0	1.125	1.25	1.375	1.5	1.625	1.75	1.875

Table 33-12. CONTRAST parameter

Figure 33-12 shows the image change depending on the contrast change. The following figure is originally intended to explain better about the contrast function as well as to show the effect of contrast reduction. Actually, the Video layer can increase the contrast, but not reduce it.

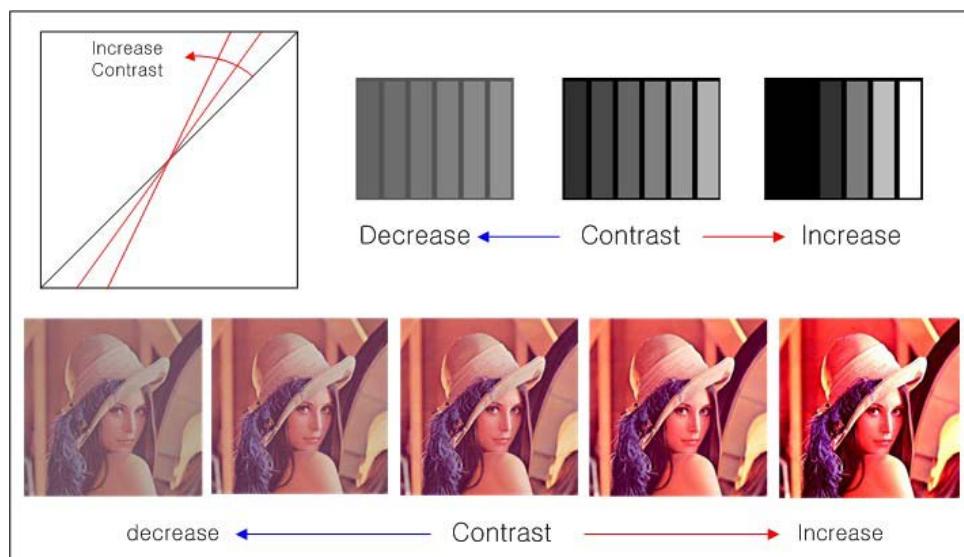


Figure 33-12. Contrast

33.4.8.2 Chrominance Enhancement

The Video layer can compensate Chrominance data by adjusting Hue and Saturation.

The Hue is adjusted by the following formulae:

$$(B-Y)' = (B-Y) * \cos(\theta) - (R-Y) * \sin(\theta)$$

$$(R-Y)' = (B-Y) * \sin(\theta) + (R-Y) * \cos(\theta)$$

The Saturation can be adjusted by multiplying a gain value by the above result value.

The Video layer can adjust the Hue and Saturation differently in each quadrant and has **HUECBnA/B** and **HUECRnA/B** parameters for each quadrant. Each parameter has the [S.1.6] format and is applied as follows:

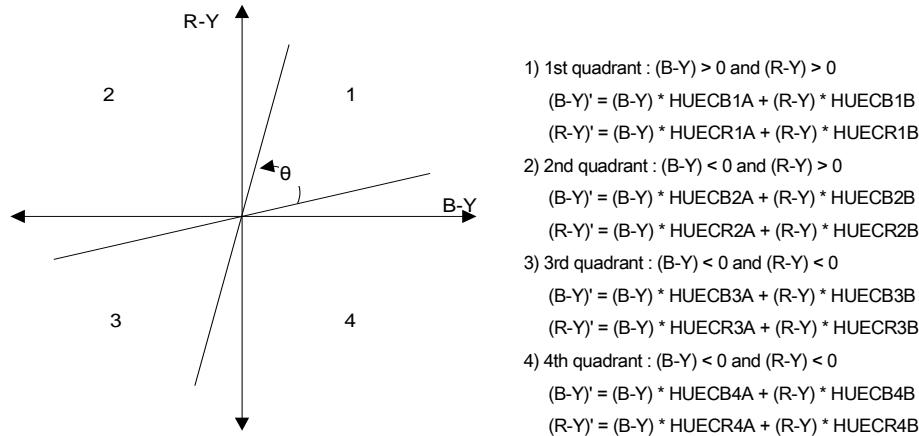


Figure 33-13. The basic concept of Hue and Saturation control

Therefore, each parameter can be calculated by using the following formulas:

- ※ The formula for Hue and Saturation parameters
 - $HUECBnA = \cos(\theta) * 64 * \text{gain}$, $HUECBnB = -\sin(\theta) * 64 * \text{gain}$
 - $HUECRnA = \sin(\theta) * 64 * \text{gain}$, $HUECRnB = \cos(\theta) * 64 * \text{gain}$
- , where θ is for hue and gain is for saturation from -2 to 1.99999X.

Figure 33-14. shows the image change depending on the changes to Hue and Saturation.

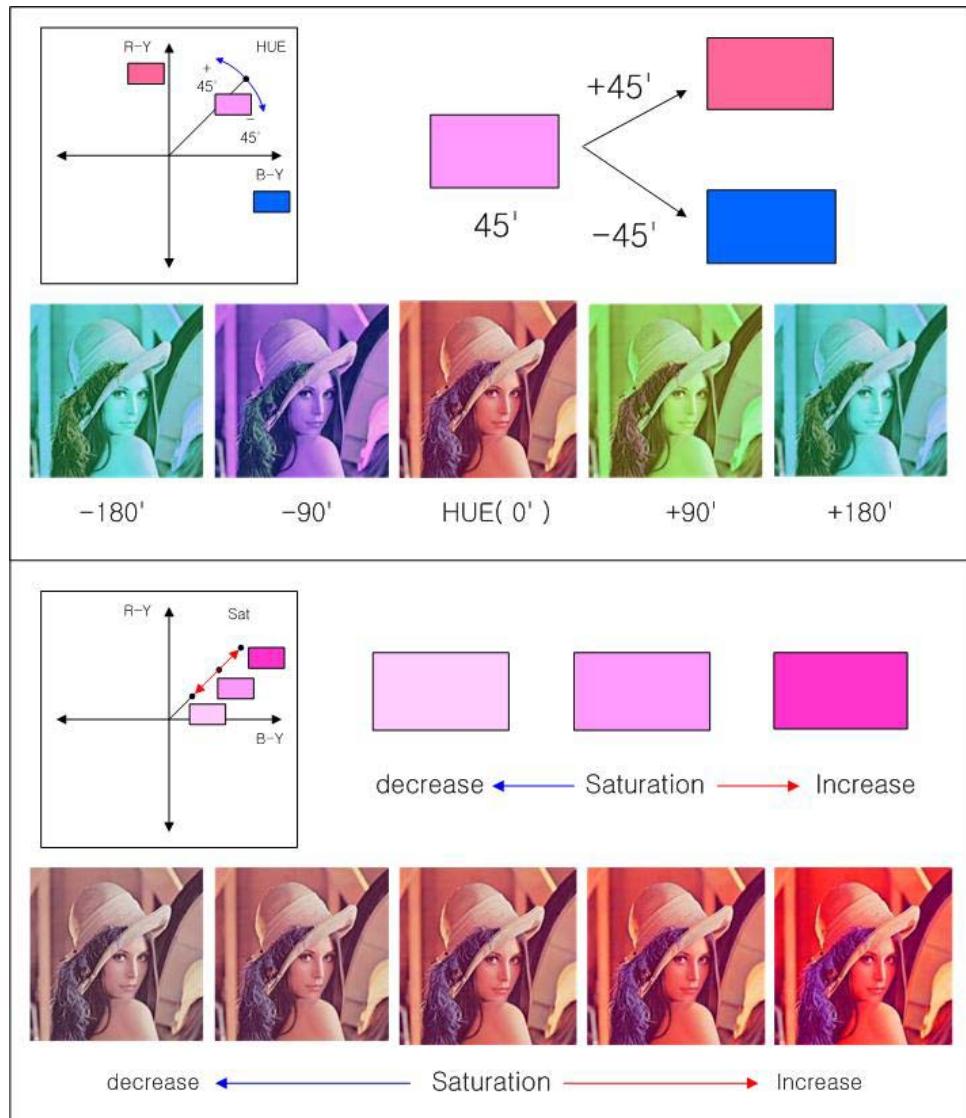


Figure 33-15. Hue and Saturation

33.4.9 Gamma Correction

MLC has three Gamma Table(256x10bit) for Red, Green, and Blue colors. Gamma[9:0] consists of Gamma[9:2], which are the integers(0~255), and Gamma[1:0], which are decimals(0.0, 0.5, 0.25, 0.75). And, R10, G10, B10, as the result of Gamma correction, are transformed into R8, G8, B8, which then can be used as input pixel for Display block , and it is possible to apply dithering while 10bit results is being transformed into 8bit..

Gamma correction can choose any Gamma region according to 3 different modes(total layer, RGB layer, Video layer), and also can designate the Alpha-blended regions of Video layer and RGB layer as the region of RGB layer or Video layer.

```

Gamma Table R = (255 * (R / 255) ^ ) << 2
Gamma Table G = (255 * (G / 255) ^ ) << 2
Gamma Table B = (255 * (B / 255) ^ ) << 2

```

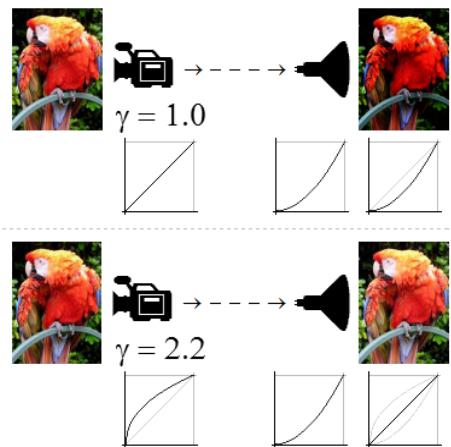


Figure 33-16. Gamma correction

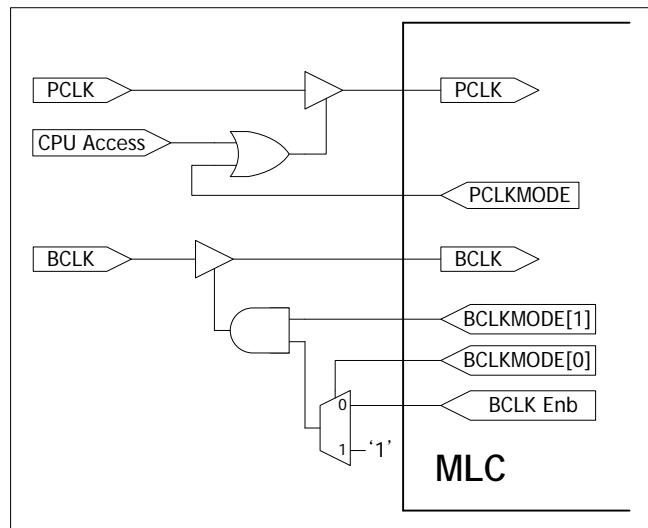
Register Set in case of Gamma correction

Ex)

- R,G,B Gamma Table Power On (MLCGAMMACONT Register R/G/BGAMMATABLE_PWD bit = '1')
- Gamma Table SLEEP Mode Disable (MLCGAMMACONT Register R/G/BGAMMATABLE_SLD bit = '1')
- Write the value of Gamma Table
- (Table address: MLCR/G/BGAMMATABLEREWRITE[31:24], table data: MLCR/G/BGAMMATABLEREWRITE[9:0])
- Gamma enable(Refer to MLCGAMMACONT Register)
- MLC Enable

33.5 Clock Generation

The MLC operates by using the PCLK and the BCLK. The PCLK is used when the CPU accesses the registers of the MLC. The BCLK is used as an internal clock or when the MLC accesses the memory. The MLC provides various operation modes for the PCLK and the BCLK. Therefore, users can adjust the clock for the MLC by setting the **PCLKMODE** and the **BCLKMODE** parameters according to their purpose. Users must set Always Mode for using the MLC. (BCLK and PCLK both)



33.6 Register Summary

Bit	R/W	Symbol	Description	Reset Value
			2's complement and from -2048 to 2047.	
MLC RGB Layer 0 top bottom Register (MLCTOPBOTTOM0)				
Address : C010_2010h/C010_2410h				
[31:28]	R	RESERVED	Reserved	4'b0
[27:16]	R/W	TOP	Specifies the y-coordinate of the upper-left corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
[15:12]	R	RESERVED	Reserved	4'b0
[11:0]	R/W	BOTTOM	Specifies the y-coordinate of the lower-right corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
MLC RGB Layer 0 INVALID AREA 0 LEFT right Register (MLCLEFTRIGHT0_0)				
Address : C010_2014h/C010_2414h				
[31:29]	R	RESERVED	Reserved	3'b0
[28]	R/W	INVALIDENB0	Shows the status of disable/enable about 1 st invisible area of RGB Layer0. 0 : Disable 1 : Enable	1'b0
[27]	R	RESERVED	Reserved	1'b0
[26:16]	R/W	LEFT	Notify left coordination of X axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	RIGHT	Notify right coordination of X axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0
MLC RGB Layer 0 INVALID AREA 0 bottom top Register (MLCTOPBOTTOM0_0)				
Address : C010_2018h/C010_2418h				
[31:27]	R	RESERVED	Reserved	5'b0
[26:16]	R/W	TOP	Notify upper coordination of Y axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	BOTTOM	Notify lower coordination of Y axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0
MLC RGB Layer 0 INVALID AREA 1 LEFT right Register (MLCLEFTRIGHT0_1)				
Address : C010_201Ch/C010_241Ch				
[31:29]	R	RESERVED	Reserved	3'b0
[28]	R/W	INVALIDENB1	Shows the status of disable/enable about 2 nd invisible area of RGB Layer0. 0 : Disable 1 : Enable	1'b0
[27]	R	RESERVED	Reserved	1'b0
[26:16]	R/W	LEFT	Notify left coordination of X axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	RIGHT	Notify right coordination of X axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0
MLC RGB Layer 0 INVALID AREA 1 bottom top Register (MLCTOPBOTTOM0_1)				
Address : C010_2020h/C010_2420h				

Bit	R/W	Symbol	Description	Reset Value
[31:27]	R	RESERVED	Reserved	5'b0
[26:16]	R/W	TOP	Notify upper coordination of Y axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	BOTTOM	Notify lower coordination of Y axis from the area to be invisible of RGB Layer0. Able to set coordination from 0 to 2047.	11'b0

MLC RGB Layer 0 control Register (MLCCONTROL0)

Address : C010_2024h / C010_2424h

[31:16]	R/W	FORMAT	Specifies the RGB data format. For detailed information, refer to Table 33-5.	16'b0
[15:14]	R	RESERVED	Reserved	2'b0
[13:12]	R/W	LOCKSIZE	Adjusts the data size to be read at any one time when a memory read for RGB data is requested. For a resolution of 1280 x 1024, it is recommended to set it as 16. 0 : 4 1 : 8 2 : 16 3 : reserved	2'b0
[11:9]	R	RESERVED	Reserved	3'b0
[8]	R/W	RESERVED	Reserved but it should be written '0'	1'b0
[7:6]	R	RESERVED	Reserved	2'b0
[5]	R/W	LAYERENB	Enables/disables this layer. 0 : Disable 1 : Enable	1'b0
[4]	R/W	DIRTYFLAG	Dirty flag for this layer. If this bit is set as '1', the register settings concerned with this layer are updated in vertical sync mode and this bit is cleared as '0' automatically. Read) 0 : Clean 1 : Dirty Write) 0 : No affect 1 : Apply modified settings	1'b0
[3]	R	RESERVED	Reserved	1'b0
[2]	R/W	BLENDENB	Enables/disables the Alpha Blending function in this layer. In addition, this bit should be set as 'enable' to apply Alpha to a format with an Alpha channel. 0 : Disable 1 : Enable	1'b0
[1]	R/W	INVENB	Enables/disables the Color Inversion function in this layer. 0 : Disable 1 : Enable	1'b0
[0]	R/W	TPENB	Enables/disables the Transparency function in this layer. 0 : Disable 1 : Enable	1'b0

MLC RGB Layer 0 HORIZONTAL stride Register (MLCHSTRIDE0)

Address : C010_2028h/C010_2428h

[31]	R/W	RESERVED	Reserved for future use. You have to write '0' only.	1'b0
[30:0]	R/W	HSTRIDE	Specifies the horizontal stride in byte units. This value is the byte offset from the current pixel to the next unit and has the number of bytes per pixel in general.	31'b0

MLC RGB Layer 0 VERTICAL stride Register (MLCVSTRIDE0)

Address : C010_202Ch/C010_242Ch

[31:0]	R/W	VSTRIDE	Specifies the vertical stride in byte units. This value is the byte offset from the current line to the next line and has the number of bytes per line in general. This value has the 2's complement format and can be specified as a negative number for the Vertical Flip function.	32'b0
--------	-----	---------	---	-------

MLC RGB Layer 0 TRANSPARENCY Color Register (MLCTPCOLOR0)

Address : C010_2030h/C010_2430h

[31:24]	R/W	ALPHA	Specifies an Alpha Blending factor. This value is valid only for an RGB format without Alpha channels. The formula for Alpha Blending is as follows: If ALPHA is 0 then a is 0, else a is ALPHA + 1. color = this layer color * a / 256 + lower layer color * (256-a) / 256	8'b0
[23:0]	R/W	TPCOLOR	Specifies the color for the Transparency. These bits have the R8G8B8 format in 24 bpp mode.	24'b0

Bit	R/W	Symbol	Description	Reset Value
MLC RGB Layer 0 InvERSION Color Register (MLCINVCOLOR0)				
<i>Address : C010_2034h/C010_2434h</i>				
[31:24]	R	RESERVED	Reserved	8'b0
[23:0]	R/W	INVCOLOR	Specifies the color to be inverted. These bits have the R8G8B8 format in 24 bpp mode.	24'b0
MLC RGB Layer 0 BASE Address Register (MLCADDRESS0)				
<i>Address : C010_2038h/C010_2438h</i>				
[31:0]	R/W	ADDRESS	Specifies the memory address where RGB data is stored. In general, the address on the top left of the image is specified, but the address on the bottom left corner is specified for Vertical Flip.	32'b0
RESERVED				
<i>Address : C010_203Ch/C010_243Ch</i>				
MLC RGB Layer 1 left right Register (MLCLEFTRIGHT1)				
<i>Address : C010_2040h</i>				
[31:28]	R	RESERVED	Reserved	4'b0
[11:0]	R/W	LEFT	Specifies the x-coordinate of the upper-left corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
[15:12]	R	RESERVED	Reserved	4'b0
[27:16]	R/W	RIGHT	Specifies the x-coordinate of the lower-right corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
MLC RGB Layer 1 top bottom Register (MLCTOPBOTTOM1)				
<i>Address : C010_2044h</i>				
[31:28]	R	RESERVED	Reserved	4'b0
[27:16]	R/W	TOP	Specifies the y-coordinate of the upper-left corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
[15:12]	R	RESERVED	Reserved	4'b0
[11:0]	R/W	BOTTOM	Specifies the y-coordinate of the lower-right corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
MLC RGB Layer 1 INVALID AREA 0 LEFT right Register (MLCLEFTRIGHT1_0)				
<i>Address : C010_2048h</i>				
[31:29]	R	RESERVED	Reserved	3'b0
[28]	R/W	INVALIDENB0	Shows the status of disable/enable about 1 st invisible area of RGB Layer1. 0 : Disable 1 : Enable	1'b0
[27]	R	RESERVED	Reserved	1'b0
[26:16]	R/W	LEFT	Notify left coordination of X axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	RIGHT	Notify right coordination of X axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
MLC RGB Layer 1 INVALID AREA 0 bottom top Register (MLCTOPBOTTOM1_0)				
<i>Address : C010_204Ch</i>				

Bit	R/W	Symbol	Description	Reset Value
[31:27]	R	RESERVED	Reserved	5'b0
[26:16]	R/W	TOP	Notify upper coordination of Y axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	BOTTOM	Notify lower coordination of Y axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
MLC RGB Layer 1 INVALID AREA 1 LEFT right Register (MLCLEFTRIGHT1_1)				
Address : C010_2050h				
[31:29]	R	RESERVED	Reserved	3'b0
[28]	R/W	INVALIDENB1	Shows the status of disable/enable about 2 nd invisible area of RGB Layer1. 0 : Disable 1 : Enable	1'b0
[27]	R	RESERVED	Reserved	1'b0
[26:16]	R/W	LEFT	Notify left coordination of X axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	RIGHT	Notify right coordination of X axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
MLC RGB Layer 1 INVALID AREA 1 bottom top Register (MLCTOPBOTTOM1_1)				
Address : C010_2054h				
[31:27]	R	RESERVED	Reserved	5'b0
[26:16]	R/W	TOP	Notify upper coordination of Y axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
[15:11]	R	RESERVED	Reserved	5'b0
[10:0]	R/W	BOTTOM	Notify lower coordination of Y axis from the area to be invisible of RGB Layer1. Able to set coordination from 0 to 2047.	11'b0
MLC RGB Layer 1 control Register (MLCCONTROL1)				
Address : C010_2058h				
[31:16]	R/W	FORMAT	Specifies the RGB data format. For detailed information, refer to Table 33-5.	16'b0
[15:14]	R	RESERVED	Reserved	2'b0
[13:12]	R/W	LOCKSIZE	Adjusts the data size to be read at any one time when a memory read for RGB data is requested. For a resolution of 1280 x 1024, it is recommended to set it as 16. 0 : 4 1 : 8 2 : 16 3 : reserved	2'b0
[11:9]	R	RESERVED	Reserved	3'b0
[8]	R/W	RESERVED	Reserved but you have to write '0' only	1'b0
[7:6]	R	RESERVED	Reserved	2'b0
[5]	R/W	LAYERENB	Enables/disables this layer. 0 : Disable 1 : Enable	1'b0
[4]	R/W	DIRTYFLAG	Dirty flag for this layer. If this bit is set as '1', the register settings concerned with this layer are updated in vertical sync mode and this bit is cleared as '0' automatically. Read) 0 : Clean 1 : Dirty Write) 0 : No affect 1 : Apply modified settings	1'b0
[3]	R	RESERVED	Reserved	1'b0
[2]	R/W	BLENDENB	Enables/disables the Alpha Blending function in this layer. In addition, this bit should be set as	1'b0

Bit	R/W	Symbol	Description	Reset Value
[27:16]	R/W	TOP	Specifies the y-coordinate of the upper-left corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
[15:12]	R	RESERVED	Reserved	4'b0
[11:0]	R/W	BOTTOM	Specifies the y-coordinate of the lower-right corner of the layer. This value has 2's complement and from -2048 to 2047.	12'b0
MLC VIDEO Layer control Register (MLCCONTROL2)				
Address : C010_207Ch / C010_247Ch				
[31:19]	R	RESERVED	Reserved	16'b0
[18:16]	R/W	FORMAT	Specifies the YUV data format. 0 : 2D block addressing separated YUV 420 (each component has 8 bit data width) 1 : 2D block addressing separated YUV 422 (each component has 8 bit data width) 2 : Linear YUV 422 (YUYV) 3 : 2D block addressing separated YUV 444 (each component has 8 bit data width) 4 : 2D block addressing separated Y/UV 422 (each component has 8 bit data width) 5 : 2D block addressing separated Y/UV 420 (each component has 8 bit data width)	2'b00
[15]	R/W	LIENBUFFER_PWD	Video Layer Line Buffer's Power On/Off 0 : Power Off 1 : Power On	1'b0
[14]	R/W	LIENBUFFER_SLMD	Video Layer Line Buffer's Sleep Mode. It is usable only when lienbuffer_pwd = '1' 0 : Sleep Mode Enable 1 : Sleep Mode Disable	1'b0
[5]	R/W	LAYERENB	Enables/disables this layer. 0 : Disable 1 : Enable	1'b0
[4]	R/W	DIRTYFLAG	Dirty flag for this layer. If this bit is set as '1', the register settings concerned with this layer are updated in vertical sync mode and this bit is cleared as '0' automatically. Read) 0 : Clean 1 : Dirty Write) 0 : No affect 1 : Apply modified settings	1'b0
[3]	R	RESERVED	Reserved	1'b0
[2]	R/W	BLENDENB	Enables/disables the Alpha Blending function in this layer. 0 : Disable 1 : Enable	1'b0
[1:0]	R/W	RESERVED	Reserved for future use. You have to write '0' only.	2'b0
MLC VIDEO Layer VERTICAL stride Register (MLCVSTRIDE3)				
Address : C010_2080h / C010_2480h				
[31:0]	R/W	VSTRIDE	Specifies the vertical stride in byte units. This value is the byte offset from the current line to the next line and has the number of bytes per line in general. In the 2D block addressing separated YUV format, this value only corresponds to the Y component and should be set as 4096 in general.	32'b0
MLC VIDEO Layer TRANSPARENCY Color Register (MLCTPCOLOR3)				
Address : C010_2084h / C010_2484h				
[31:24]	R/W	ALPHA	Specifies an Alpha Blending factor. The formula for Alpha Blending is as follows: If ALPHA is 0 then a is 0, else a is ALPHA + 1. color = this layer color * a / 256 + lower layer color * (256-a) / 256	8'b0
[23:0]	R	RESERVED	Reserved	24'b0
RESERVED				
Address : C010_2088h / C010_2488h				
[31:0]	-	RESERVED	Reserved	-

Bit	R/W	Symbol	Description	Reset Value
MLC VIDEO Layer BASE Address Register (MLCADDRESS3)				
Address : C010_208Ch / C010_248Ch				
[31:0]	R/W	ADDRESS	Specifies the memory address where YUB data is stored. In the 2D block addressing separated YUV format, this value only corresponds to the Y component and should be set as follows: • ADDRESS[31:24] : the index of segment • ADDRESS[23:12] : y-coordinate in segment • ADDRESS[11:0] : x-coordinate in segment, ADDRESS[2:0] must be '0'	32'b0
MLC VIDEO Layer Cb BASE Address Register (MLCADDRESSCB)				
Address : C010_2090h / C010_2490h				
[31:0]	R/W	ADDRESSCB	Specifies the memory address where Cb data is stored. These bits are valid only for the 2D block addressing separated YUV format and should be set as follows: • ADDRESS[31:24] : the index of segment • ADDRESS[23:12] : y-coordinate in segment • ADDRESS[11:0] : x-coordinate in segment, ADDRESS[2:0] must be '0'	32'b0
MLC VIDEO Layer Cr BASE Address Register (MLCADDRESSCR)				
Address : C010_2094h / C010_2494h				
[31:0]	R/W	ADDRESSCR	Specifies the memory address where Cr data is stored. These bits are valid only for the 2D block addressing separated YUV format and should be set as follows: • ADDRESS[31:24] : the index of segment • ADDRESS[23:12] : y-coordinate in segment • ADDRESS[11:0] : x-coordinate in segment, ADDRESS[2:0] must be '0'	32'b0
MLC VIDEO Layer Cb VERTICAL stride Register (MLCVSTRIDECB)				
Address : C010_2098h / C010_2498h				
[31:0]	R/W	VSTRIDECB	Specifies the vertical stride for Cr data in byte units. These bits are valid only for the 2D block addressing separated YUV format and should be set as 4096 in general.	32'b0
MLC VIDEO Layer Cr VERTICAL stride Register (MLCVSTRIDEKR)				
Address : C010_209Ch / C010_249Ch				
[31:0]	R/W	VSTRIDEKR	Specifies the vertical stride for Cr data in byte units. These bits are valid only for the 2D block addressing separated YUV format and should be set as 4096 in general.	32'b0
MLC VIDEO Layer Horizontal Scale Register (MLCHScALE)				
Address : C010_20A0h / C010_24A0h				
[31:30]	R	RESERVED	Reserved	2'b0
[29]	R/W	HFILTERENB_C	Decides whether to use bilinear filter when Video Layer horizontal scale.(Chroma filter enable) 0 : Disable (point sample) 1 : Enable (bilinear filter)	1'b0
[28]	R/W	HFILTERENB	Decides whether to use bilinear filter when Video Layer horizontal scale.(Luminance filter enable) 0 : Disable (point sample) 1 : Enable (bilinear filter)	1'b0
[27:23]	R	RESERVED	Reserved	5'b0
[22:0]	R/W	HSCALE	Specifies the ratio for the horizontal scale. The formula to calculate this value is as follows: When FILTERENB is 1 and the destination width is wider than the source width: • HSCALE = (source width-1) * (1<<11) / (destination width-1) , else • HSCALE = source width * (1<<11) / destination width	23'h800
MLC VIDEO Layer Vertical Scale Register (MLCVScALE)				
Address : C010_20A4h / C010_24A4h				
[31:30]	R	RESERVED	Reserved	2'b0
[29]	R/W	VFILTERENB_C	Decides whether to use bilinear filter when Video Layer vertical scale.(Chroma filter enable) 0 : Disable (Nearest sample) 1 : Enable (bilinear filter)	1'b0

Bit	R/W	Symbol	Description	Reset Value
[28]	R/W	VFILTERENB	Decides whether to use bilinear filter when Video Layer vertical scale.(Luminance filter enable) 0 : Disable (Nearest sample) 1 : Enable (bilinear filter)	1'b0
[27:23]	R	RESERVED	Reserved	5'b0
[22:0]	R/W	VSCALE	Specifies the ratio for the vertical scale. The formula to calculate this value is as follows: When FILTERENB is 1, the destination height is higher than the source height: • VSCALE = (source height-1) * (1<<11) / (destination height-1) , else • VSCALE = source height * (1<<11) / destination height	23'h800

MLC VIDEO Layer Luminanace Enhancement Control Register (MLCLUENH)

Address : C010_20A8h/C010_24A8h

[31:16]	R	RESERVED	Reserved	16'b0
[15:8]	R/W	BRIGHTNESS	Specifies brightness values in 256 levels. These values are 2's complements and can be set between -128 and +127.	8'b0
[7:3]	R	RESERVED	Reserved	5'b0
[2:0]	R/W	CONTRAST	Specifies contrast levels with 8 levels. 0 : 1.0 1 : 1.125 2 : 1.25 3 : 1.375 4 : 1.5 5 : 1.625 6 : 1.75 7 : 1.875	3'b0

MLC VIDEO Layer Chrominiance Enhancement Control 0 Register (MLCCHENH0)

Address : C010_20ACh/C010_24ACh

[31:24]	R/W	HUECR1B	Specifies the factors for Hue and Saturation for the first quadrant. Each value has the 2's complement format and the format is [S.1.6] (here, S means a sign bit).	8'h40
[23:16]	R/W	HUECR1A	The Hues and Saturations for each quadrant are determined by the following formulae: • (B-Y)' = (B-Y) * HUECBnA + (R-Y) * HUECBnB • (R-Y)' = (B-Y) * HUECRnA + (R-Y) * HUECRnB	8'b0
[15:8]	R/W	HUECB1B		8'b0
[7:0]	R/W	HUECB1A	The formulae for each factor are as follows: • HUECBnA = cos(θ) * 64 * gain • HUECBnB = -sin(θ) * 64 * gain • HUECRnA = sin(θ) * 64 * gain • HUECRnB = cos(θ) * 64 * gain where the gain value is between 0 and 2.	8'h40

MLC VIDEO Layer Chrominiance Enhancement Control 1 Register (MLCCHENH1)

Address : C010_20B0h/C010_24B0h

[31:24]	R/W	HUECR2B	Specifies the factors for Hue and Saturation for the second quadrant. Each value has the 2's complement format and the format is [S.1.6] (here, S means a sign bit).	8'h40
[23:16]	R/W	HUECR2A	The Hues and Saturations for each quadrant are determined by the following formulae: • (B-Y)' = (B-Y) * HUECBnA + (R-Y) * HUECBnB • (R-Y)' = (B-Y) * HUECRnA + (R-Y) * HUECRnB	8'b0
[15:8]	R/W	HUECB2B		8'b0
[7:0]	R/W	HUECB2A	The formulae for each factor are as follows: • HUECBnA = cos(θ) * 64 * gain • HUECBnB = -sin(θ) * 64 * gain • HUECRnA = sin(θ) * 64 * gain • HUECRnB = cos(θ) * 64 * gain where the gain value is between 0 and 2.	8'h40

MLC VIDEO Layer Chrominiance Enhancement Control 2 Register (MLCCHENH2)

Address : C010_20B4h/C010_24B4h

[31:24]	R/W	HUECR3B	Specifies the factors for Hue and Saturation for the third quadrant. Each value has the 2's complement format and the format is [S.1.6] (here, S means a sign bit).	8'h40
[23:16]	R/W	HUECR3A	The Hues and Saturations for each quadrant are determined by the following formulae: • (B-Y)' = (B-Y) * HUECBnA + (R-Y) * HUECBnB • (R-Y)' = (B-Y) * HUECRnA + (R-Y) * HUECRnB	8'b0
[15:8]	R/W	HUECB3B		8'b0
[7:0]	R/W	HUECB3A	The formulae for each factor are as follows: • HUECBnA = cos(θ) * 64 * gain • HUECBnB = -sin(θ) * 64 * gain • HUECRnA = sin(θ) * 64 * gain • HUECRnB = cos(θ) * 64 * gain where the gain value is between 0 and 2.	8'h40

MLC VIDEO Layer Chrominiance Enhancement Control 3 Register (MLCCHENH3)

Bit	R/W	Symbol	Description	Reset Value
Address : C010_20B8h / C010_24B8h				
[31:24]	R/W	HUECR4B	Specifies the factors for Hue and Saturation for the fourth quadrant. Each value has the 2's complement format and the format is [S.16] (here, S means a sign bit).	8'h40
[23:16]	R/W	HUECR4A	The Hues and Saturations for each quadrant are determined by the following formulae: • (B-Y)' = (B-Y) * HUECBnA + (R-Y) * HUECBnB • (R-Y)' = (B-Y) * HUECRnA + (R-Y) * HUECRnB	8'b0
[15:8]	R/W	HUECB4B	The formulae for each factor are as follows: • HUECBnA = cos(θ) * 64 * gain • HUECBnB = -sin(θ) * 64 * gain • HUECRnA = sin(θ) * 64 * gain • HUECRnB = cos(θ) * 64 * gain where the gain value is between 0 and 2.	8'b0
[7:0]	R/W	HUECB4A		8'h40
MLC GAMMA CONTROL Register (MLCGAMMACONT)				
Address : C010_20ECh / C010_24ECh				
[31:12]	R	RESERVED	Reserved	19'b0
[11]	R/W	BGAMMATABL_E_PWD	'Blue' Gamma Table Power On/Off it should be 'On' before MLC GAMMA Enabled. 0 : Power Off 1 : Power On	1'b0
[10]	R/W	BGAMMATABL_E_SLD	'Blue' Gamma Table Sleep Mode it is usable only when Power On. 0 : Sleep Mode Enable 1 : Sleep Mode Disable	1'b0
[9]	R/W	GGAMMATABL_E_PWD	'Green' Gamma Table Power On/Off it should be 'On' before MLC GAMMA Enabled. 0 : Power Off 1 : Power On	1'b0
[8]	R/W	GGAMMATABL_E_SLD	'Green' Gamma Table Sleep Mode it is usable only when Power On. 0 : Sleep Mode Enable 1 : Sleep Mode Disable	1'b0
[7:6]	R	RESERVED	Reserved	2'b0
[5]	R/W	ALPHASELECT	Determine whether the Alpha blended region with RGB layer, in Video layer, should be processed as Video layer or as RGB layer. 0 : RGB 1 : Video	1'b0
[4]	R/W	YUVGAMMAENB	Gamma Enable for the Video region 0 : Disable 1 : Enable	1'b0
[3]	R/W	RGAMMATABL_E_PWD	'Red' Gamma Table Power On/Off it should be 'On' before MLC GAMMA Enabled. 0 : Power Off 1 : Power On	1'b0
[2]	R/W	RGAMMATABL_E_SLD	'Red' Gamma Table Sleep Mode it is usable only when Power On. 0 : Sleep Mode Enable 1 : Sleep Mode Disable	1'b0
[1]	R/W	RGBGAMMAEMB	Gamma Enable for the RGB region 0 : Disable 1 : Enable	1'b0
[0]	R/W	DITHERENB	Dither Enable Enables/disables the dithering operation when 10bit, as the result of Gamma correction, is transformed to 8bit. 0 : Disable 1 : Enable	1'b0
MLC RED GAMMA TABLE WRITE(MLCRGAMMATABLEREWRITE)				
Address : C010_20F0h / C010_24F0h				
[31:24]	W	TABLEADDR	Table Write Address(Size : 10bit x 256)	-
[23:10]	-	RESERVED	Reserved	-
[9:0]	W	TABLEDATA	Table Write Data	-
MLC GREEN GAMMA TABLE WRITE(MLCGGAMMATABLEREWRITE)				
Address : C010_20F4h / C010_24F4h				

Bit	R/W	Symbol	Description	Reset Value
[31:24]	W	TABLEADDR	Table Write Address	-
[23:10]	-	RESERVED	Reserved	-
[9:0]	W	TABLEDATA	Table Write Data	-
MLC BLUE GAMMA TABLE WRITE(MLCBGAMMATABLEWRITE)				
Address : C010_20F8h/C010_24F8h				
[31:24]	W	TABLEADDR	Table Write Address	-
[23:10]	-	RESERVED	Reserved	-
[9:0]	W	TABLEDATA	Table Write Data	-
RESERVED				
Address : C010_20FCh~C010_23BC/C010_24FCh~C010_27BC				
MLC Clock Generation Enable Register (MLCCLKENB)				
Address : C010_23C0h/C010_27C0h				
[31:24]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0 : PCLK is always disabled (can access CLKGEN only) 1 : PCLK is always enabled	1'b0
[2]	R	RESERVED	Reserved	1'b0
[1:0]	R/W	BCLKMODE	Specifies BCLK operating mode. 0 : BCLK is always disabled 1 : Reserved (Never use this) 2 : Reserved (Never use this) 3 : BCLK is always enabled	2'b0

Section 34. Display Controller (DPC)

34.1 Overview

The Display controller (hereinafter, DPC) is a block that generates the signals to interface with external display devices, such as a TFT-LCD, or video encoder. The DPC consists of a Sync generator. The Sync generator transmits control signals to the Multi-Layer Controller (MLC) and receives RGB data from the MLC. Then the Sync generator converts the received RGB data into a suitable format. In addition, the Sync generator can supports various types of LCD and video encoders adjusting various output formats and Sync signals.

34.1.1 Features

- Supports RGB, Multiplexed RGB (MRGB), ITU-R BT.601, ITU-R BT.656 and MPU Type (i80)
- Programmable HSYNC, VSYNC and DE (Data Enable)
- Programmable polarity for Sync signals
- Programmable delay for data, Sync signals and clock phase
- Supports UPScaler (Only Secondary Display)
- Supports RGB dithering
- VCLK(video clock) max frequency 150MHz
- Max resolution 2048x2048

34.1.2 Block Diagram

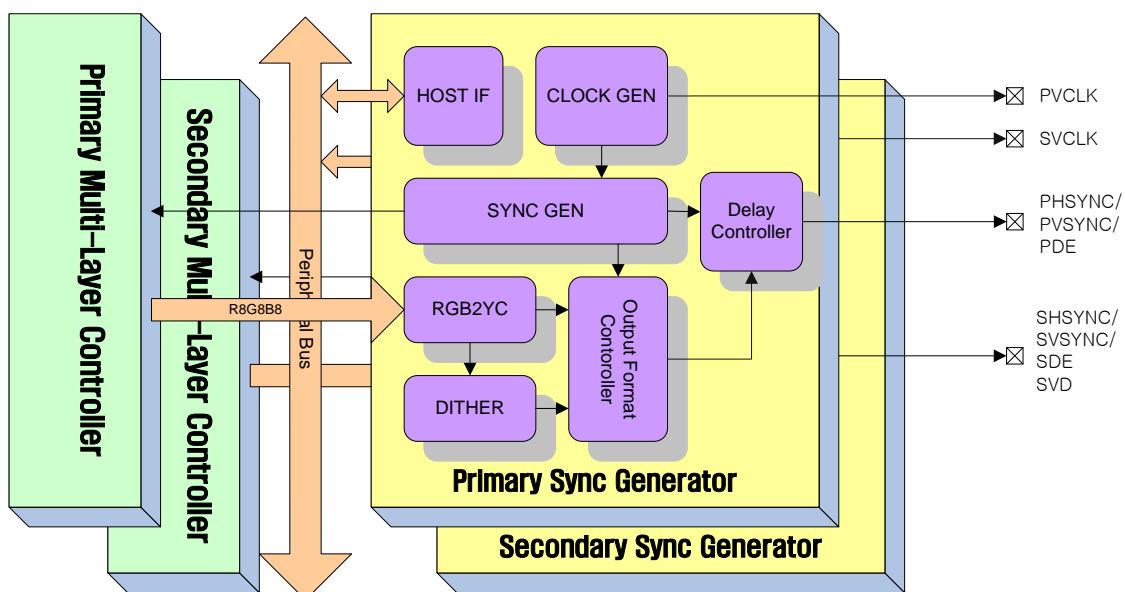


Figure 34-1. Display Controller

34.2 Sync Generator

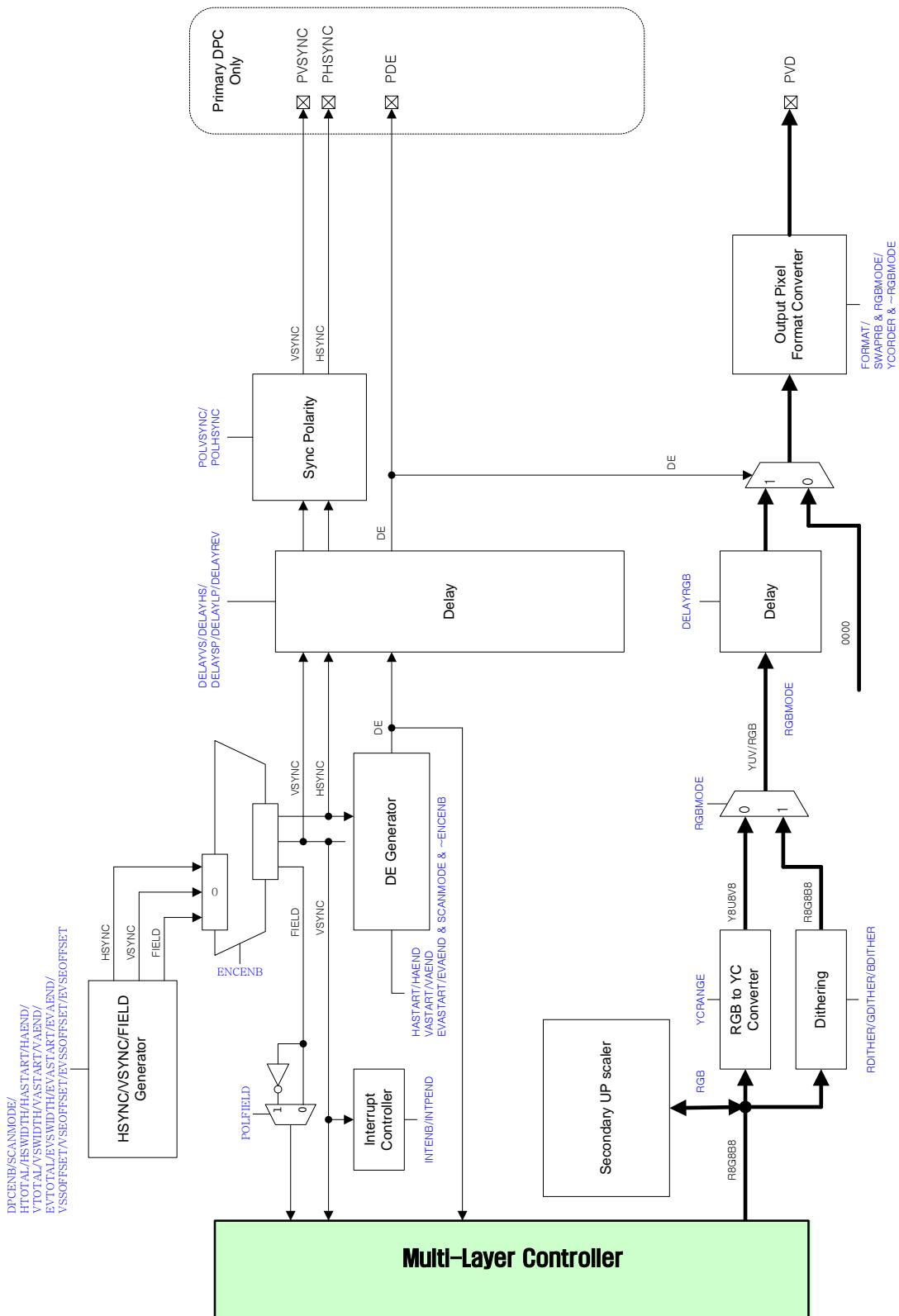


Figure 34-2. Sync Generator

34.2.1 Clock generation

Peripheral Clock Generation

The PCLK is used when the CPU accesses the registers of the DPC. Users can adjust the DPC clock by setting the *PCLKMODE* for the user's purpose.

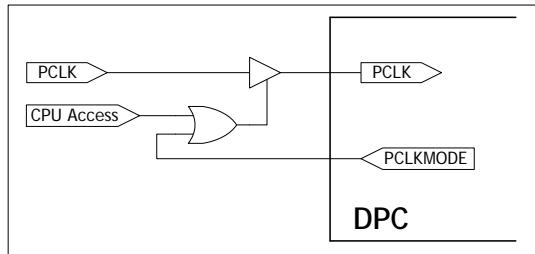


Figure 34-3. Peripheral Clock Generation

PCLKMODE	Brief Description
0	Always Disable
1	Always enabled..

Table 34-1. PCLK mode

Video Clock Generation

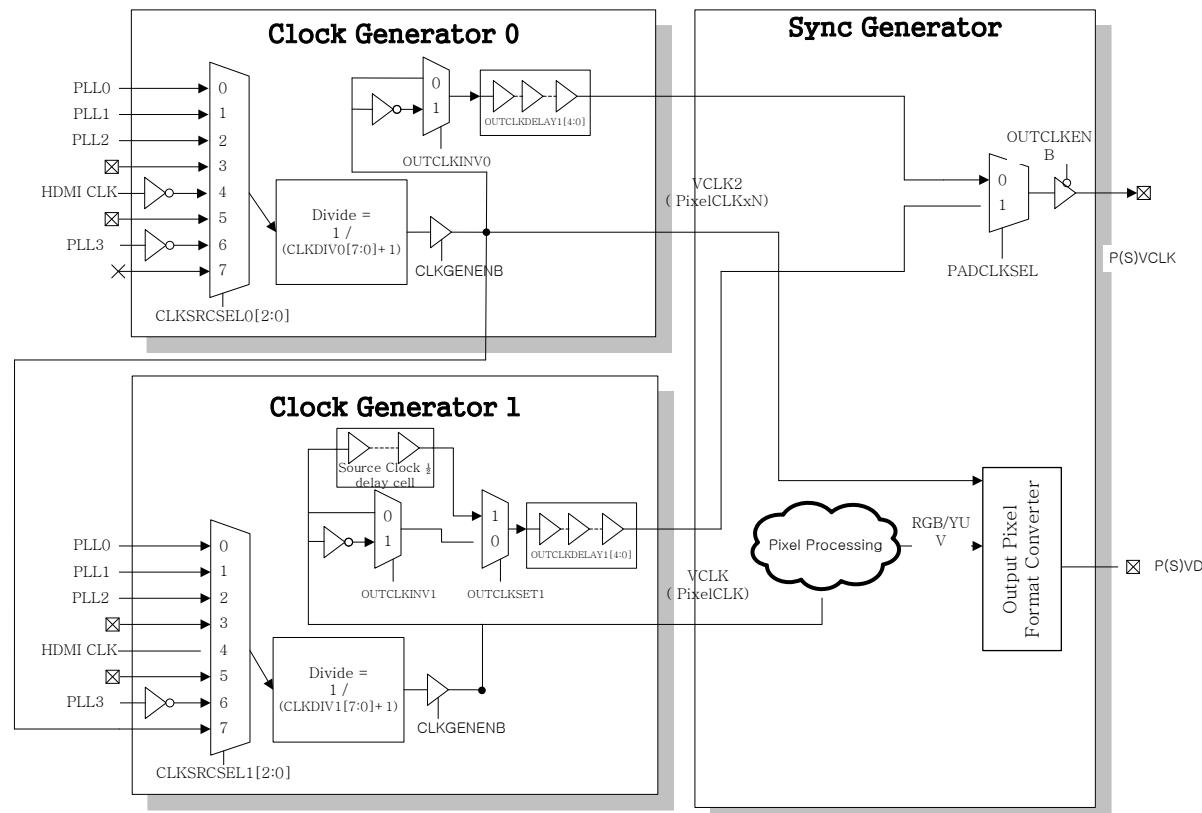


Figure 34-4. Video Clock Generation

The DPC can create an output clock via various clock sources. The clocks for creating an output clock are 2-PLL, PADCLK(P/SCLK, ACLK), etc.

The sync generator uses the VCLK and the VCLK2 as clock sources. The VCLK is a clock for the operation of a pixel unit. The VCLK2 is a clock for pixel output. Therefore, the VCLK is one clock per pixel and the VCLK2 is 1~6 clocks per pixel depending on the output format. In the RGB and ITU-R BT.601A formats, which output one pixel data per clock, the VCLK and the VCLK2 share the same clock. In the MRGB, the ITU-R BT.601B, the ITU-R BT.601(8 bit) and the ITU-R BT.656 formats, which output one pixel data per two clocks, the VCLK should divide the VCLK2 by 2. Since the VCLK should divide the VCLK2 by 2, Clock Generator1 should use the output of Clock Generator0 as the clock source. Users can select the output source by using the **PADCLKSEL**. The selection of the **PADCLKSEL** is not related to the operation of the sync generator. In general, VCLK2 is used as the output clock. If, however, a display device using a dual edge is connected, the VCLK is used as the output clock.

Format	CLKSRCSEL0 [2:0]	CLKDIV0 [7:0]	CLKSRCSEL1 [2:0]	CLKDIV1 [5:0]	OUTCLKSEL1	PADCLKSEL
RGB, ITU-R BT.601A	0~6	0~256	7	0	0	1
MRGB, ITU-R BT.601B, 601(8bit), 656, MPU (i80)	0~6	0~256	7	1	0	1
MRGB (Dual edge)	0~6	0~256	7	1	1	0
SRGB888	0~6	0~256	7	6	0	2
SRGBD8888	0~6	0~256	7	4	0	1

Table 34-2. Recommend Clock Settings

The DPC can adjust the polarity and phase of the output clock. The **OUTCLKINV** adjusts the polarity of the output clock and the **OUTCLKDELAY**, **OUTCLKSEL** adjusts the phase of the output clock.

Basically, the DPC outputs data to be fetched at the falling edge and the **OUTCLKINV** is set as '0'. For a display device that fetches the clock at the rising edge, the **OUTCLKINV** should be set as '1' to invert the output clock.

OUTCLKDELAY	0	1	2	3	4	5	6	7
tDelay (ns)	TBD							

Table 34-3. Clock delay

The output clock is changed by the **OUTCLKINV** and the **OUTCLKDELAY** as shown in Figure 34-5.

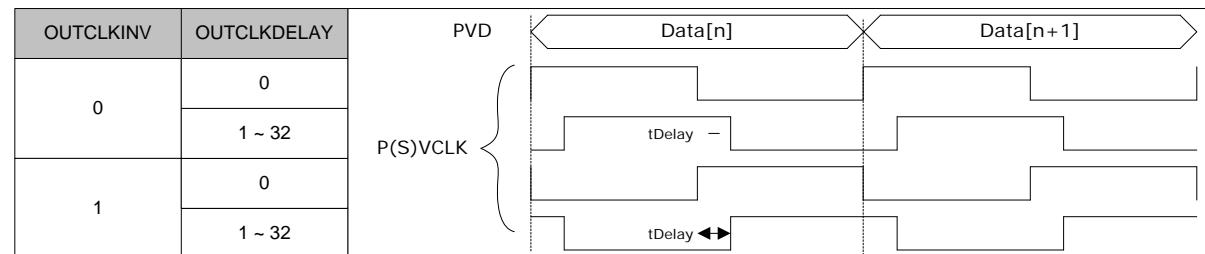


Figure 34-5. Clock polarity and delay

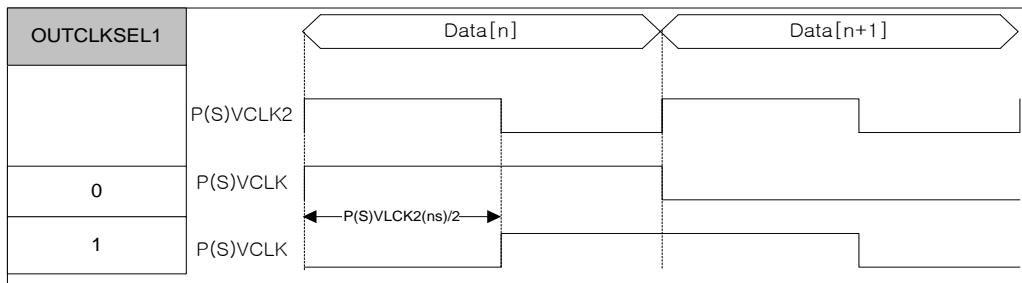


Figure 34-6. OUTCLK delay

34.2.2 Format

The sync generator can receive RGB888 data from the MLC and display the data in various formats.

The formats that can be displayed by the primary sync generator are listed in Table 34-4.

Output format	RGB MODE	FORMAT	CLK	PVD																										
				23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RGB555	1	0	-									R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0				
RGB565		1	-	R4	R3	R2	R1	R0				G5	G4	G3	G2	G1	G0		B4	B3	B2	B1	B0							
RGB666		2	-	R5	R4	R3	R2	R1	R0			G5	G4	G3	G2	G1	G0		B5	B4	B3	B2	B1	B0						
RGB888		3	-	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0			
MRGB555A		4	1 st																G2	G1	G0	B4	B3	B2	B1	B0				
			2 nd																R4	R3	R2	R1	R0	G4	G3					
MRGB555B		5	1 st																G2	G1	G0	B4	B3	B2	B1	B0				
			2 nd																R4	R3	R2	R1	R0	G4	G3					
MRGB565		6	1 st																G2	G1	G0	B4	B3	B2	B1	B0				
			2 nd																R4	R3	R2	R1	R0	G5	G4	G3				
MRGB666	2	7	1 st																G2	G1	G0	B5	B4	B3	B2	B1	B0			
			2 nd																R5	R4	R3	R2	R1	R0	G5	G4	G3			
MRGB888A		8	1 st																G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
			2 nd																R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4
MRGB888B	0	9	1 st																G4	G3	G2	B7	B6	B5	B4	B3	G0	B2	B1	B0
			2 nd																R7	R6	R5	R4	R3	G7	G6	G5	R2	R1	R0	G1
ITU-R BT.656 BT.601(8bit)		10	1 st										Cb7	Cb6	Cb5	Cb4	Cb3	Cb2	Cb1	Cb0										
			2 nd										Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0										
			3 rd										Cr7	Cr6	Cr5	Cr4	Cr3	Cr2	Cr1	Cr0										
			4 th										Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0										
ITU-R BT.601A		12	1 st										Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cb7	Cb6	Cb5	Cb4	Cb3	Cb2	Cb1	Cb0		
			2 nd										Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cr7	Cr6	Cr5	Cr4	Cr3	Cr2	Cr1	Cr0		
ITU-R BT.601B		13	1 st										Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cb7	Cb6	Cb5	Cb4	Cb3	Cb2	Cb1	Cb0		
			2 nd										Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cr7	Cr6	Cr5	Cr4	Cr3	Cr2	Cr1	Cr0		
4096 Color	X	1	-																R0	G0	B0	R1	B1	G1	R2	G2				

16 Level Gray	X	3	-	16 gray bars	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
---------------	---	---	---	--------------	----	----	----	----	----	----	----	----

Table 34-4. Data format for Primary Sync Generator

Up to 24-bit data is available in the primary sync generator. If, however, the display format is not RGB666 or RGB888, the higher 8-bit is not used.

RGB Format

In the RGB format, data are displayed in the order of blue components, green components and red components based on the lower data. However, a user can swap the display of red components and blue components by setting the **SWAPRB** as '1'.

In addition, the DPC supports the Dithering effect in RGB format. All RGB data transmitted from the MLC have 8-bit data width. Therefore, if the data width of each component is less than 8-bit, as in RGB565 and RGB666 display, the lower bits are discarded. In such cases, the display quality can be compensated by the Dithering effect.

When RGB images are displayed as RGB565, the dithered image shows the difference from the image displayed, as shown in Figure 34-7, after the unused lower bits are simply discarded.

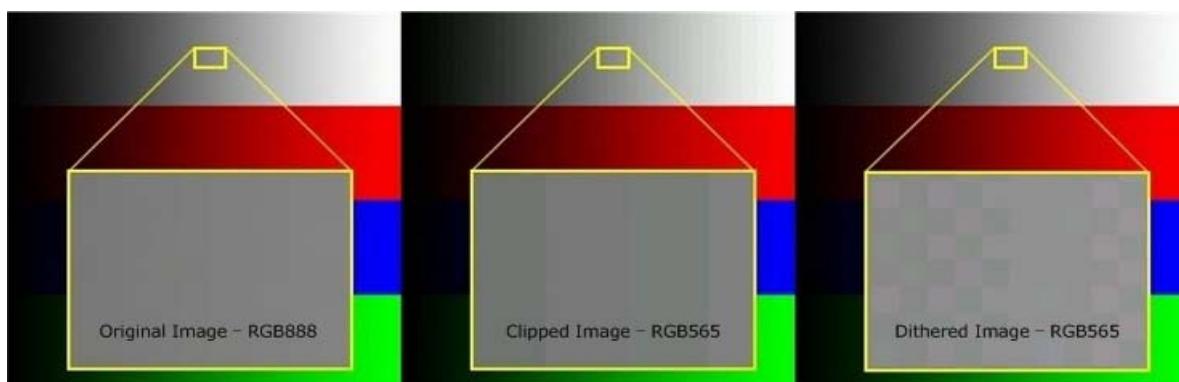


Figure 34-7. RGB Dithering

The RGB Dithering is set by **RDITHER**, **GDITHER** and **BDITHER** and the setting values by output formats are listed in Table 34-5.

Output Format	RDITHER	GDITHER	BDITHER
RGB555, MRGB555A, MRGB555B	5 bit Dither	5 bit Dither	5 bit Dither
RGB565, MRGB565	5 bit Dither	6 bit Dither	5 bit Dither
RGB666, MRGB666	6 bit Dither	6 bit Dither	6 bit Dither
STN - 4096 Color, 16 Level Gray	4 bit Dither	4 bit Dither	4 bit Dither
RGB888, MRGB888A/B, ITU-R BT.656, ITU-R BT.601A/B	Bypass	Bypass	Bypass

Table 34-5. Recommend Setting for RGB Dithering

YCbCr Format

Since the DPC only receives RGB data from the MLC, it outputs the data after converting the RGB data to YCbCr data for the ITU-R BT.656 display or the ITU-R BT.601 display. The DPC converts RGB data to YCbCr data by using the following formulae:

*The formula for RGB to YCbCr conversion

- $Y = 0.229 * R + 0.587 * G + 0.114 * B$
- $Cb = -0.169 * R - 0.331 * G + 0.5 * B$
- $Cr = 0.5 * R - 0.419 * G - 0.081 * B$

When ITU-R BT.601 B external display device is used, the user can change the data output order by adjusting the **YCORDER**. The output data for **YCORDER** can be changed as listed in Table 34-6.

YCORDER	0		1	
CLK	1st	2nd	1st	2nd
VD[15]	Y[7]		Y[7]	
VD[14]	Y[6]		Y[6]	
VD[13]	Y[5]		Y[5]	
VD[12]	Y[4]		Y[4]	
VD[11]	Y[3]		Y[3]	
VD[10]	Y[2]		Y[2]	
VD[9]	Y[1]		Y[1]	
VD[8]	Y[0]		Y[0]	
VD[7]	Cr[7]	Cb[7]	Cb[7]	Cr[7]
VD[6]	Cr[6]	Cb[6]	Cb[6]	Cr[6]
VD[5]	Cr[5]	Cb[5]	Cb[5]	Cr[5]
VD[4]	Cr[4]	Cb[4]	Cb[4]	Cr[4]
VD[3]	Cr[3]	Cb[3]	Cb[3]	Cr[3]
VD[2]	Cr[2]	Cb[2]	Cb[2]	Cr[2]
VD[1]	Cr[1]	Cb[1]	Cb[1]	Cr[1]
VD[0]	Cr[0]	Cb[0]	Cb[0]	Cr[0]

Table 34-6. Output order for ITU-R BT. 601 B

34.2.3 Sync signals

The sync generator creates sync signals with various timings. The primary sync generator transmits HSYNC, VSYNC and DE signals to the outside to provide timing interfaces for external display devices. Users can program each sync signal setting to create the timings required from external display devices.

Horizontal Timing Interface

HSYNC and DE signals are used for Horizontal Sync. The horizontal timing consists of tHSW, tHBP, tHFPP and tAVW as shown in Figure 34-8.

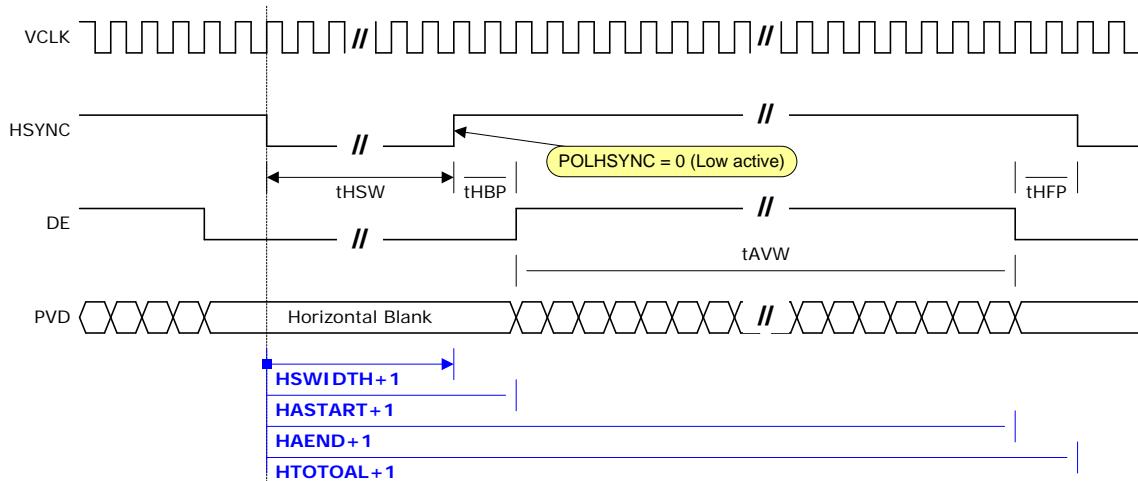


Figure 34-8. Horizontal timing

Each symbol in the above figure is described in Table 34-7.

Symbol	Brief	Remark
tHSW	Horizontal Sync Width	Number of VCLKs in the section where the horizontal sync pulse is active
tHBP	Horizontal Back Porch	Number of VCLKs in a section from the end point of the horizontal sync pulse to the start point of the horizontal active
tHFP	Horizontal Front Porch	Number of VCLKs in a section from the end point of the horizontal active to the start point of the horizontal sync pulse
tAVW	Active Video Width	Number of VCLKs in a horizontal active section

Table 34-7. Horizontal timing symbols

The horizontal timing setting registers are **HWIDTH**, **HSTART**, **HAEND** and **HTOTAL** and each register setting is described in Table 34-8. Each unit of the registers is based on VCLK and each value is set as 'total number - 1'.

Register	Formula	Remark
HWIDTH	$tHSW - 1$	Number of VCLKs in a section from the start point of the horizontal sync pulse to the end point of the horizontal sync pulse - 1
HSTART	$tHSW + tHBP - 1$	Number of VCLKs in a section from the start point of the horizontal sync pulse to the start point of the horizontal active - 1
HAEND	$tHSW + tHBP + tAVW - 1$	Number of VCLKs in a section from the start point of the horizontal sync pulse to the end point of the horizontal active - 1
HTOTAL	$tHSW + tHBP + tAVW + tHFP - 1$	Number of VCLKs in a section from the start point of the horizontal sync pulse to the start point of the next horizontal sync pulse - 1

Table 34-8. Horizontal timing registers

The **POLHsync** is used to change the polarity of the HSYNC signal to be output to the outside. The horizontal sync pulse is low active when the **POLHsync** is '0', while the horizontal sync pulse is high active when the **POLHsync** is '1'. The polarity of the Data Enable (DE) signal to be output to the outside cannot be changed and the section that outputs valid data comes into high state.

Vertical Timing Interface

The VSYNC signal is used for the Vertical Sync. The vertical timing consists of tVSW, tVBP, tVFP and tAVH as shown in Figure 34-9. In addition, the relation between the Vertical Sync and the Horizontal Sync is established by tVSSO and tVSEO.

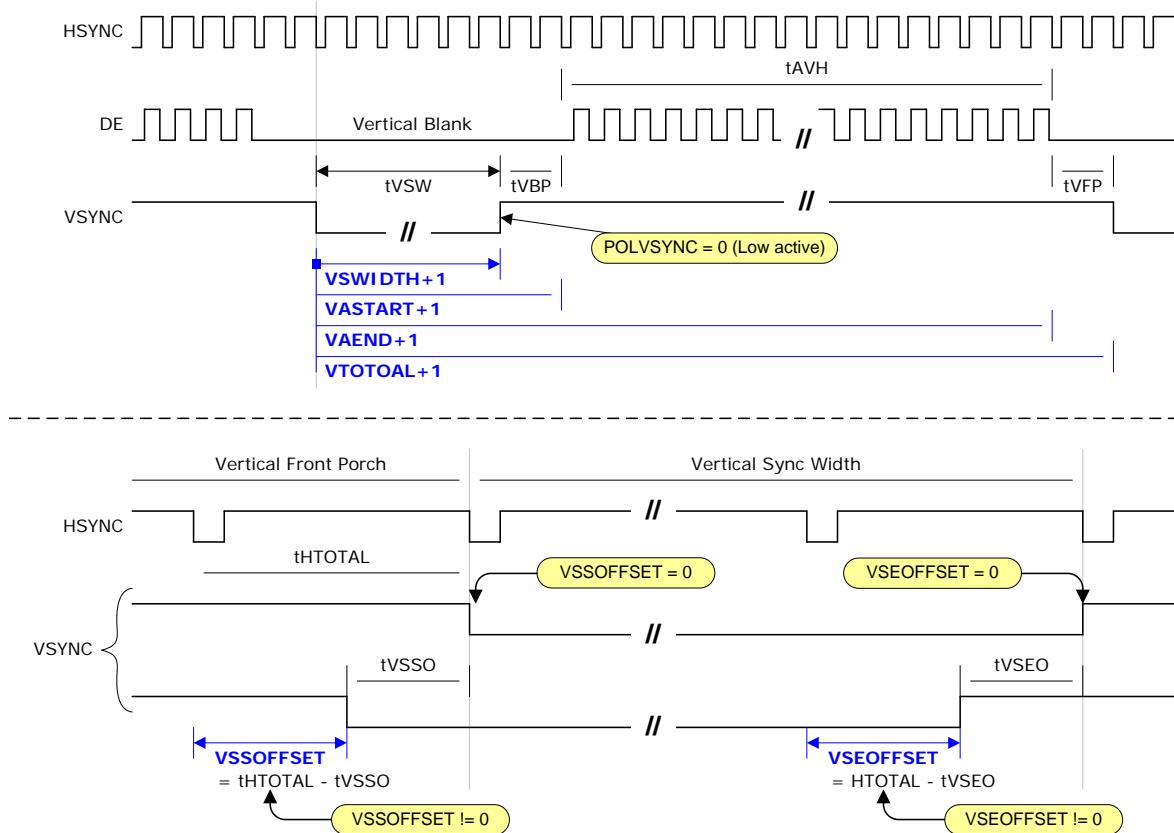


Figure 34-9. Vertical timing

Each symbol in the above figure is described in Table 34-9.

Symbol	Brief	Remark
tVSW	Vertical Sync Width	Number of lines in the section where the vertical sync pulse is active
tVBP	Vertical Back Porch	Number of lines in the section from the end point of the vertical sync pulse to the start point of the next vertical active
tVFP	Vertical Front Porch	Number of lines in the section from the end point of the vertical active to the start point of the vertical sync pulse
tAVH	Active Video Height	Number of lines in the vertical active section
tHTOTAL	Horizontal Total	Number of total VCLKs in a horizontal cycle where the horizontal active section and the horizontal blank section are added
tVSSO	Vertical Sync Start Offset	Number of VCLKs in a section from the start point of the vertical sync pulse to the start point of the horizontal sync pulse
tVSEO	Vertical Sync End Offset	Number of the VCLKs in a section from the end point of the horizontal sync pulse to the start point of the vertical sync pulse

Table 34-9. Vertical timing symbols

The vertical timing setting registers are **VSWIDTH**, **VSTART**, **VAEND** and **VTOTAL** and each register setting is described in Table 34-10. The units of **VSWIDTH**, **VSTART**, **VAEND** and **VTOTAL** are based on the horizontal lines and their values are set as 'total number - 1'. The units of **VSSOFFSET** and **VSEOFFSET** are based on the VCLK.

Register	Formula	Remark
VSWIDTH	$tVSW - 1$	Number of lines in a section from the start point of the vertical sync pulse to the end point of the vertical sync pulse - 1
VSTART	$tVSW + tVBP - 1$	Number of VCLKs in a section from the start point of the vertical sync pulse to the start point of the horizontal active section

		vertical active - 1
VAEND	$t_{VSW} + t_{VBP} + t_{AVH} - 1$	Number of lines in a section from the start point of the vertical sync pulse to the end point of the vertical active - 1
VTOTAL	$t_{VSW} + t_{VBP} + t_{AVH} + t_{VFP} - 1$	Number of lines in a section from the start point of the vertical sync pulse to the next point of the vertical sync pulse - 1
VSSOFFSET	If t_{VSSO} is 0 then 0, else $t_{HTOTAL} - t_{VSSO}$	Number of VCLKs in a section from the start point of the last horizontal sync pulse in the Vertical Front Porch to the start point of the vertical sync pulse. If, however, VSSOFFSET is equal to t_{HTOTAL} , the value is set as '0'.
VSEOFFSET	If t_{VSEO} is 0 then 0, else $t_{HTOTAL} - t_{VSEO}$	Number of VCLKs in a section from the start point of the last horizontal sync pulse in the Vertical Front Porch to the start point of the vertical sync pulse. If, however, VSEOFFSET is equal to t_{HTOTAL} , the value is set as '0'.

Table 34-10. Vertical timing registers

The **POLVSYNC** is used to change the polarity of the VSYNC signal to be output to the outside. The vertical sync pulse is low active when the **POLVSYNC** is '0', while the vertical sync pulse is high active when the **POLVSYNC** is '1'.

AC Timing

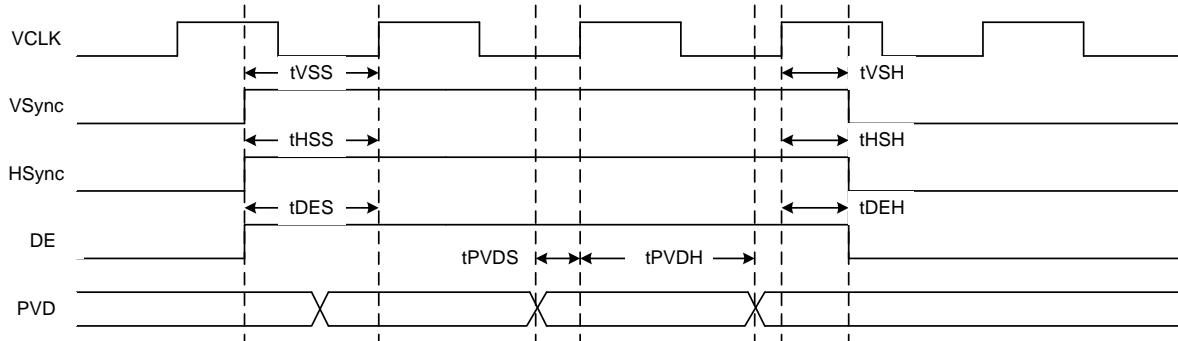


Figure 34-10. Vertical timing

Condition : VCLK 40MHz(25ns)

Symbol	Min	Max	Unit	Description
tVSS	12.13	-	ns	VSync setup time
tVSH	12.28	-	ns	VSync hold time
tHSS	12.27	-	ns	HSync setup time
tHSH	12.26	-	ns	HSync hold time
tDES	12.1	-	ns	DE setup time
tDEH	12.4	-	ns	DE hold time
tPVDS	8.8	-	ns	PVD setup time
tPVDH	12.87	-	ns	PVD hold time

Table 34-11. sync timing symbols

UPSCALER (Only Secondary Display)

UPSCALER performs the function of horizontal bilinear filter upscale MLC OUT Layer. It is selectable whether to scale with UPSCALEENB Setting.

UPSCALEENB Setting

$$\text{UPSCALE} = (\text{MLCScreenWidth}-1) * (1<<11) / (\text{destination width}-1)$$

Embedded Sync

The ITU-R BT.656 output does not need the separate Sync Signal pin to transmit signals to the outside. The Sync information is transmitted along with data via a data pin. At this time, the Sync information is inserted as a separate code before the start point of the valid data (SAV) and after the end point of the valid data (EAV). The Sync information included in data is as shown in Figure 34-11.

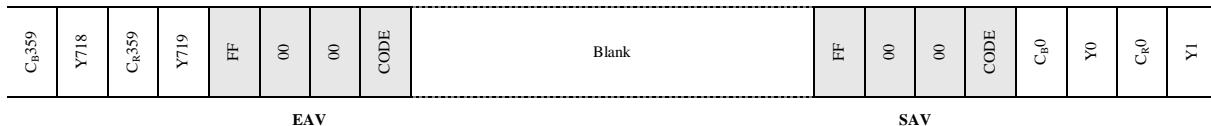


Figure 34-11. Data stream format with SAV/EAV

The SAV and EAV consist of [FF, 00, 00, CODE]. Each of the SAV and EAV codes contains Field(F), VSYNC(V) and HSYNC(H) data and each code is composed as follows:

Bit		7	6	5	4	3	2	1	0	Hex	Brief Description
Function		1	F	V	H	P3	P2	P1	P0		
(FVH)	0	1	0	0	0	0	0	0	0	80h	SAV of odd field
	1	1	0	0	1	1	1	0	1	9Dh	EAV of odd field
	2	1	0	1	0	1	0	1	1	ABh	SAV of odd blank
	3	1	0	1	1	0	1	1	0	B6h	EAV of odd blank
	4	1	1	0	0	0	1	1	1	C7h	SAV of even field
	5	1	1	0	1	1	0	1	0	DAh	EAV of even field
	6	1	1	1	0	1	1	0	0	ECh	SAV of even blank
	7	1	1	1	1	0	0	0	1	F1h	EAV of even blank
<ul style="list-style-type: none"> ▪ F : Field select (0 : odd field, 1 : even field) ▪ V : Vertical blanking (0 : Active, 1 : blank) ▪ H : SAV/EAV (0 : SAV, 1 : EAV) ▪ Parity : P3 = V xor H, P2 = F xor H, P1 = F xor V, P0 = F xor V xor H 											

Table 34-12. Embedded Sync Code

In the ITU-R BT.656 format, SAV/EAV codes are inserted in data by setting the **SEAVENB** as '1'. However, since the SAV/EAV codes are transmitted via a data pin, the range of the data should be restricted, to distinguish the codes from the data. Users can restrict the data range by using **YCRANGE**. Figure 34-12 shows that the result of the color space conversion changing RGB data to YCbCr data varies depending on YCRANGE.

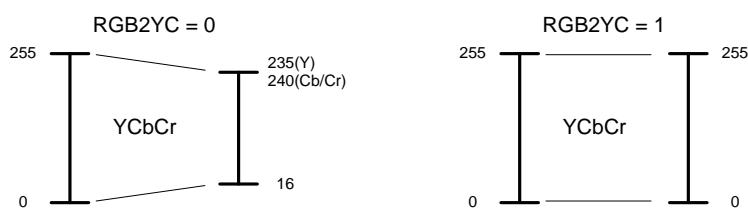


Figure 34-12. Y/C clip

If **YCRANGE** is set as '0', a Y value between 0 and 15 is changed to 16 and a Y value between 236 and 255 is changed to 235. In a similar way, Cb/Cr values between 0 and 15 are changed to 16 and Cb/Cr values between 241

and 255 are changed to 240. If **YCRANGE** is set as '1', all Y/Cb/Cr values are bypassed.

34.2.4 Scan mode

The sync generator supports interface display as well as progressive display. The Scan mode is determined by **SCANMODE**. If **SCANMODE** is '0', the sync generator operates in progressive scan mode. If **SCANMODE** is '1', the sync generator operates in interlaced scan mode. In interlaced scan mode, the different vertical timing interfaces can be set in the odd and even fields separately. The odd field display and the progressive display share same registers and registers for the even field display exists separately. Registers to create vertical timing in accordance with the scan mode is listed in Table 34-13.

SCANMODE	Scan mode		Registers
0	Progressive		
1	Interface	Odd field	VTOTAL, VSWIDTH, VASTART, VAEND, VSSOFFSET, VSEOFFSET
		Even field	EVTOTAL, EVSWIDTH, EVASTART, EVAEND, EVSSOFFSET, EVSEOFFSET

Table 34-13. Registers relative to scan mode

34.2.5 Delay

The pixel data is more delayed than the final output sync signal due to the processing in the sync generator. Therefore, users should delay Sync signal output to synchronize the Sync signals with the pixel data. When the Sync generator processes data, 4-clock is consumed for RGB data and 6-clock is consumed for YCbCr data. Therefore, for synchronization between Sync signals and data, the output of the Sync signals should be delayed. In RGB format and ITU-R BT.601A format with the same VCLK and VCLK2, delays of 4-clock and 6-clock should be set in each format, separately. In the MGRB and the ITU-R BT.656/601B formats, where the VCLK2 is twice the VCLK, delays of 8-clock and 12-clock should be set in each format, separately.

unit : VCLK2

Format	VCLK2 : VCLK	DELAYRGB	DELAYHS, DELAYVS, DELAYDE
RGB	1:1	0	Primary TFT :7 Primary STN :8 Secondary :7
MRGB	2:1	0	Primary TFT :14 Secondary :14
ITU-R BT.601A	1:1	0	6
ITU-R BT.656, ITU-R BT.601B	2:1	0	12

Table 34-14. Default delay value

34.2.6 Interrupt

The DPC can generate an interrupt whenever VSYNC occurs.

Interrupt invokes each vertical Sync when TFT LCD Progressive operation. Interrupt invokes each 16 vertical sync when STN LCD Progressive operation. Interrupt invokes each EVEN field(2 vertical sync) when interlace operation. The DPC sets **INTPEND** as '1' if VSYNC occurs and notifies the interrupt generation to the Interrupt controller if the **INTENB** is '1'. Therefore, users can acknowledge the generation of VSYNC via polling by using the status of the **INTPEND** regardless of the generation of interrupts. The **INTPEND** is cleared by writing '1' to it.

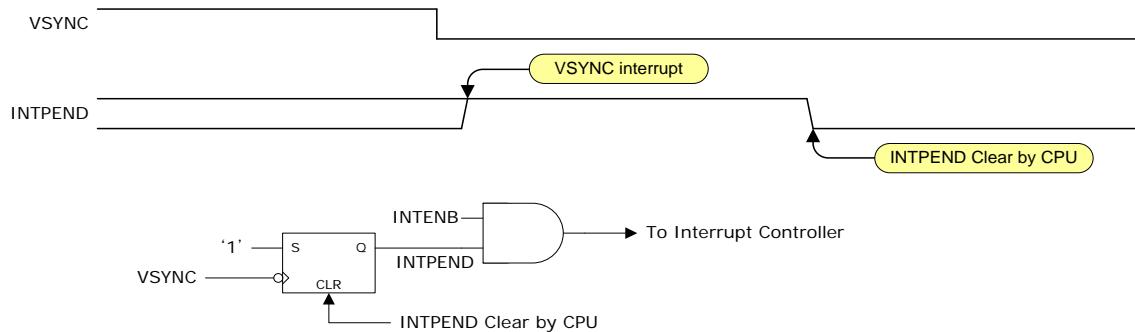


Figure 34-13. Interrupt block diagram and timing

34.2.7 MPU (i80) Type Sync signals

The sync generator can create signals for MPU LCD interface (i80). MPU LCD interface's signals are consists of **nCS** (Chip Select), **RS** (Register Select), **nWR** (Write), **nRD**(Read), and In/Out Data. MPU LCD interface has 4 type of accesses.

Signals	Access Types	Description	Register Address
RS=0 nWR	Address Write	Write index to IR	[15:0] : C010_2984 [23:16] : C010_2988
RS=0 nRD	Status Read	Read Internal Status	[23:16] : C010_298C [15:0] : C010_2994
RS=1 nWR	Data Write (or Graphic Data Write)	Write to Control Register and GRAM	[15:0] : C010_2984 [23:16] : C010_2990
RS=1 nRD	Data Read	Read from GRAM	[23:16] : C010_2990 [15:0] : C010_2994

Table 34-15. i80 Access Types

NXP4330D/Q has register for communicating with external devices, and it supports 4 types of MPU interface format. In the case of "writing", lower 16 bits need to be written to **C010_2984** address followed by writing upper 8 bits to **C010_2990**address. In the case of "reading", upper 8 bits need to be read from **C010_298C** address followed by reading lower 16 bits from **C010_2994** address. Actual communication with external devices is occurred when writing to **C010_2990**address and reading from **C010_298C** address.

To satisfy setup/hold timing of external devices, setting appropriate parameters, which include setup counter, access counter and hold counter, need to be set in DPCMPUTIME0, DPCMPUTIME1 registers. Following is the timing diagram of write operation.

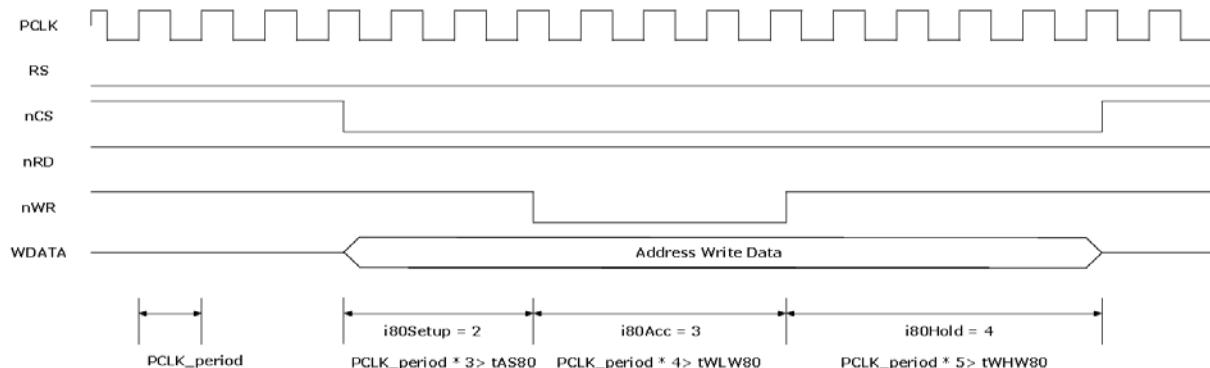


Figure 34-14. i80Address Write

NXP4330D/Q can convert video sync signals to MPU i80 type by setting DPC to i80 mode. Note that VCLK_2 needs to be 2 times faster than VCLK for MPU i80. Following diagram describes video sync to MPU i80 conversion.

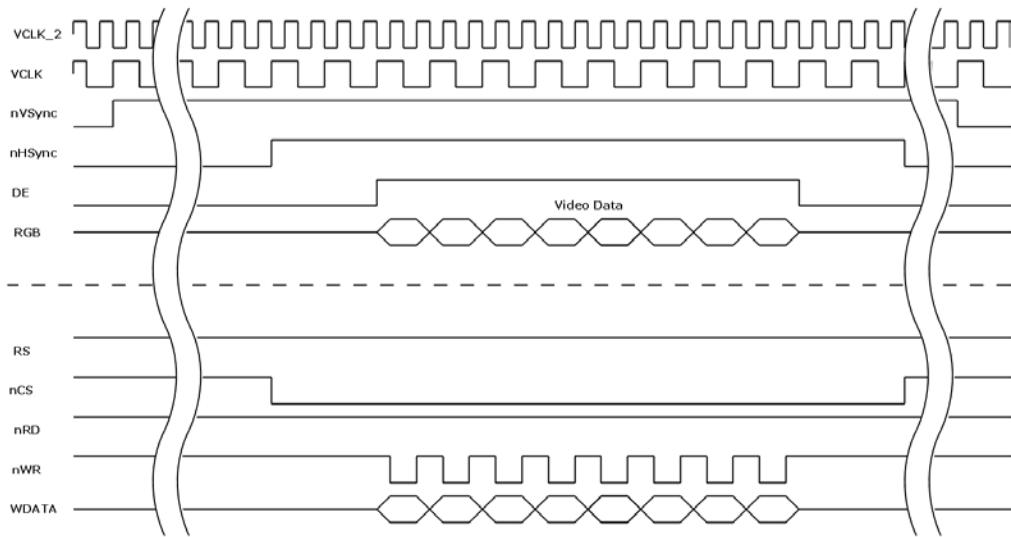


Figure 34-15. i80 Graphic Data Convert

NXP4330D/Q has a "Command Buffer", which is used to send additional information during the VSYNC period.

34.2.8 Odd/Even Field Flag

NXP4330D/Q DPC provides DPC RGB SHIFT. FIELDFLAG register which indicates whether currently being displayed frame is odd or even frame.

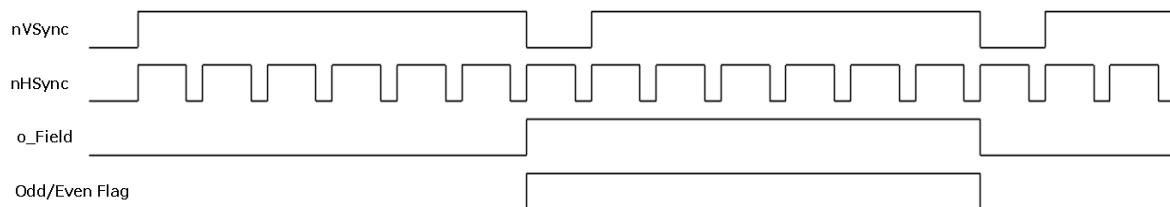


Figure 34-16. Odd/Even Flag

34.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
Reserved				
Address : C010_2800h ~ C010_28FFh/C010_2C70h ~ C010_2CFFh				
[31:0]	R	RESERVED	Reserved	32'b0
DPC Horizontal Total Length Register (DPCHTOTAL)				
Address : C010_28F8h/C010_2CF8h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	HTOTAL	Specifies the number of total VCLK clocks for a horizontal line. TFT or Video Encoder : HTOTAL = tHSW + tHBP + tHFP + tAVW - 1 Color STN : HTOTAL = CL1HW + CL1ToCL2DLY + {(tAVWx 3)/BitWidth}xCPLCYCx2 - 1 Monochrome STN : HTOTAL = CL1HW + CL1ToCL2DLY + {(tAVWx 1)/BitWidth} x CL2CYCx2 - 1	16'b0
DPC Horizontal Sync Width Register (DPCHSWIDTH)				
Address : C010_28FCh/C010_2CFCh				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	HSWIDTH	TFT or Video Encoder: Specifies the number of VCLK clocks for the horizontal sync width. This value must be less than HSTART. • HSWIDTH = tHSW - 1 STN: CL1 Height Width HSWIDTH = CL1HW - 1	16'b0
DPC Horizontal Active Video START Register (DPCHASTART)				
Address : C010_2900h/C010_2D00h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	HASTART	TFT or Video Encoder: Specifies the number of VCLK clocks from start of the horizontal sync to start of the active video. This value must be less than HAEND. • HASTART = tHSW + tHBP - 1 STN: Delay Cycle from posedge active of CL1 signal to CL2 posedge active. • HASTART = CL1ToCL2DLY	16'b0
DPC Horizontal Active Video End Register (DPCHAEND)				
Address : C010_2904h/C010_2D04h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	HAEND	TFT or Video Encoder: Specifies the number of VCLK clocks from start of the horizontal sync to end of the active video. This value must be less than HTOTAL. • HAEND = tHSW + tHBP + tAVW - 1 STN: Active Width • HAEND = tAVW - 1	16'b0
DPC Vertical Total Length Register (DPCVTOTAL)				
Address : C010_2908h/C010_2D08h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	VTOTAL	Specifies the number of total lines for a frame or field. TFT or Video Encoder : VTOTAL = tVSW + tVBP + tVFP + tAVH - 1 STN : VTOTAL = BLANKLINE + tAVH - 1	16'b0

Bit	R/W	Symbol	Description	Reset Value				
DPC Vertical Sync Width Register (DPCVSWIDTH)								
Address : C010_290Ch / C010_2D0Ch								
[31:16]	R	RESERVED	Reserved	16'b0				
[15:0]	R/W	VSWIDTH	<p>TFT or Video Encoder: Specifies the number of lines for the vertical sync width. This value must be less than VASTART. When interlace mode, this value is used for odd field.</p> <ul style="list-style-type: none"> • VSWIDTH = tVSW - 1 <p>STN : Blank Line Number • VSWIDTH = BLANKLINE - 1</p>	16'b0				
DPC Vertical Active Video START Register (DPCVASTART)								
Address : C010_2910h / C010_2D10h								
[31:16]	R	RESERVED	Reserved	16'b0				
[15:0]	R/W	VASTART	<p>Specifies the number of lines from start of the vertical sync to start of the active video. This value must be less than VAEND. When interlace mode, this value is used for odd field.</p> <ul style="list-style-type: none"> • VASTART = tVSW + tVBP - 1 	16'b0				
DPC Vertical Active Video End Register (DPCVAEND)								
Address : C010_2914h / C010_2D14h								
[31:16]	R	RESERVED	Reserved	16'b0				
[15:0]	R/W	VAEND	<p>Specifies the number of lines from start of the vertical sync to end of the active video. This value must be less than VTOTAL. When interlace mode, this value is used for odd field.</p> <ul style="list-style-type: none"> • VAEND = tVSW + tVBP + tAVH - 1 	16'b0				
DPC Control 0 Register (DPCCTRL0)								
Address : C010_2918h / C010_2D18h								
[31:16]	R	RESERVED	Reserved	16'b0				
[15]	R/W	DPCENB	<p>Enable/Disable the display controller(DPC).</p> <table style="margin-left: 20px;"> <tr> <td>0 : Disable</td> <td>1 : Enable</td> </tr> </table>	0 : Disable	1 : Enable	1b0		
0 : Disable	1 : Enable							
[14:13]	R/W	RESERVED	Reserved	2'b0				
[12]	R/W	RGBMODE	<p>Specifies the output pixel format.</p> <table style="margin-left: 20px;"> <tr> <td>0 : YCbCr1</td> <td>1 : RGB</td> </tr> </table>	0 : YCbCr1	1 : RGB	1b0		
0 : YCbCr1	1 : RGB							
[11]	R/W	INTENB	<p>Enable/Disable the VSYNC interrupt. The VSYNC interrupt will be issued at start of the VSYNC pulse. Therefore an interrupt will be occurred at every frame in progressive mode or at every field in interface mode.</p> <table style="margin-left: 20px;"> <tr> <td>0 : Disable</td> <td>1 : Enable</td> </tr> </table>	0 : Disable	1 : Enable	1b0		
0 : Disable	1 : Enable							
[10]	R/W	INTPEND	<p>Indicates whether the VSYNC interrupt is pended or not. This bit is always operated regardless of INTENB bit.</p> <table style="margin-left: 20px;"> <tr> <td>Read> 0 : Not pended</td> <td>1 : Pended</td> </tr> <tr> <td>Write> 0 : No affect</td> <td>1 : Clear</td> </tr> </table>	Read> 0 : Not pended	1 : Pended	Write> 0 : No affect	1 : Clear	1b0
Read> 0 : Not pended	1 : Pended							
Write> 0 : No affect	1 : Clear							
[9]	R/W	SCANMODE	<p>Determines whether scan mode is progressive or interlace.</p> <table style="margin-left: 20px;"> <tr> <td>0 : Progressive scan mode</td> <td>1 : Interlaced scan mode</td> </tr> </table>	0 : Progressive scan mode	1 : Interlaced scan mode	1b0		
0 : Progressive scan mode	1 : Interlaced scan mode							
[8]	R/W	SEAVENB	<p>Enable/Disable SAV/EAV signal into the data. This is used for ITU-R BT.656 format.</p> <table style="margin-left: 20px;"> <tr> <td>0 : Disable embedded sync</td> <td>1 : Enable embedded sync</td> </tr> </table>	0 : Disable embedded sync	1 : Enable embedded sync	1b0		
0 : Disable embedded sync	1 : Enable embedded sync							
[7:4]	R/W	DELAYRGB	Specifies the delay for RGB PAD output. The unit is VCLK2. Generally this value has '0' for normal operation.	4b0				
[3]	R	RESERVED	Reserved	1b0				
[2]	R/W	POLFIELD	Specifies the polarity of the internal field signal.	1b0				
[1]	R/W	POLVSYNC	Specifies the polarity of the vertical sync output. This bit is only valid in case of primary display controller.	1b0				

Bit	R/W	Symbol	Description	Reset Value
			0 : Low active 1 : High active	
[0]	R/W	POLHsync	Specifies the polarity of the horizontal sync output. This bit is only valid in case of primary display controller. 0 : Low active 1 : High active	1b0
DPC Control 1 Register (DPCCTRL1)				
Address : C010_291Ch / C010_2D1Ch				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	SWAPRB	Swap Red and Blue component. This value is only valid when the output is RGB. This bit is only valid in case of primary display controller. 0 : RGB 1 : BGR	1b0
[14]	R	RESERVED	Reserved	1b0
[13]	R/W	YCRANGE	Determines the YUV range for RGB to YUV conversion. Write 0 set of Video Encoder output 0 : Y = 16 ~ 235, Cb/Cr = 16 ~ 2401 : Y/Cb/Cr = 0 ~ 255	1b0
[12]	R	RESERVED	Reserved	1b0
[11:8]	R/W	FORMAT	Specifies the data output format. TFT or Video Encoder : 0 : RGB555 1 : RGB565 2 : RGB666 3 : RGB888 4 : MRGB555A 5 : MRGB555B 6 : MRGB565 7 : MRGB666 8 : MRGB888A 9 : MRGB888B 10 : ITU-R BT.656 or 601(8bit) 11 : Reserved 12 : ITU-R BT.601A 13 : ITU-R BT.601B(set YCORDER bit as '1') 14 : Reserved 15 : Reserved STN : 0 : Reserved 1 : 4096 Color 2 : Reserved 3 : 16 Gray Level other : Reserved	4b0
[7]	R/W	RESERVED	Reserved but this bit should be set to '0'	1b0
[6]	R/W	YCORDER	Specifies the data output order in case of ITU-R BT. 601B. 0 : Cb Y Cr Y 1 : Cr Y Cb Y	1b0
[5:4]	R/W	BDİTHER	Specifies the dithering method of Blue component. This value is only valid in case of primary display controller. 0 : Bypass 1 : 4bit dither 2 : 5bit dither 3 : 6bit dither	2b0
[3:2]	R/W	GDİTHER	Specifies the dithering method of Green component. This value is only valid in case of primary display controller. 0 : Bypass 1 : 4bit dither 2 : 5bit dither 3 : 6bit dither	2b0
[1:0]	R/W	RDİTHER	Specifies the dithering method of Red component. This value is only valid in case of primary display controller. 0 : Bypass 1 : 4bit dither 2 : 5bit dither 3 : 6bit dither	2b0
DPC Even Field Vertical Total Length Register (DPCEVTOTAL)				
Address : C010_2920h / C010_2D20h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	EVTOTAL	Specifies the number of total lines for even field. This register is only used when interface mode. • $EVTOTAL = tEVSW + tEVBP + tEVFP + tEAHV - 1$	16'b0
DPC Even Field Vertical Sync Width Register (DPCEVSWIDTH)				
Address : C010_2924h / C010_2D24h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	EVSWIDTH	Specifies the number of lines for the vertical sync width of even field. This value must be less than EVASTART. This register is only used when interface mode. • $EVSWIDTH = tEVSW - 1$	16'b0

Bit	R/W	Symbol	Description	Reset Value
DPC Even Field Vertical Active Video START Register (DPCEVASTART)				
Address : C010_2928h / C010_2D28h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	EVASTART	<p>Specifies the number of lines from start of the vertical sync to start of the active video when even field. This value must be less than EVAEND. This register is only used when interlace mode.</p> <ul style="list-style-type: none"> • EVASTART = tEVSW + tEVBP - 1 	16'b0
DPC Even Field Vertical Active Video End Register (DPCEVAEND)				
Address : C010_292Ch / C010_2D2Ch				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	EVAEND	<p>Specifies the number of lines from start of the vertical sync to end of the active video when even field. This value must be less than EVTOTAL. This register is only used when interlace mode.</p> <ul style="list-style-type: none"> • EVAEND = tEVSW + tEVBP + tEAHV - 1 	16'b0
DPC Control 2 Register (DPCCTRL2)				
Address : C010_2930h / C010_2D30h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:12]	R/W	CL2CYC	Sets STN LCD CL2 (Shift clock) Cycle (Unit : VCLK) CL2CYC = CL2CYC - 1	4'b0
[11:10]	R	RESERVED	Reserved	2'b0
[9]	R/W	STNLCDBITWIDTH	<p>STN LCD Data Bus Bit Width. 0 : Reserved 1 : 8 bit Scan mode is applicable with Dual view mode set.</p>	1'b0
[8:7]	R/W	LCDTYPE	<p>Declares External Display Device Type 0 : TFT or Video Encoder 1 : STN LCD 2 : Dual View mode(TFT or Video Encoder) 3 : Reserved</p>	2'b0
[6:5]	R	RESERVED	Reserved	2'b0
[4]	R/W	I80ENABLE	Enable MPU i80 Mode. 0: No i80 Mode 1 : i80 Mode Enable	1'b0
[3]	R	RESERVED	Reserved	1'b0
[2]	R/W	INITPADCLK	Initial Value of VCLK2 div 2. (for SRGB888)	1'b0
[1:0]	R/W	PADCLKSEL	Specifies the PAD output clock. 0 : VCLK 1 : VCLK2 2 : VCLK2 div 2 (for SRGB888) 3 : Reserved	2'b0
DPC Vertical Sync End Offset Register (DPCVSEOFFSET)				
Address : C010_2934h / C010_2D34h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	VSEOFFSET	<p>Specifies the number of clocks from start of the horizontal sync to end of the vertical sync, where the horizontal sync is last one in the vertical sync. If this value is 0 then end of the vertical sync synchronizes with start of the horizontal sync which is new one in the vertical back porch. This value has to be less than HTOTAL. When interface mode, this value is used for odd field.</p> <ul style="list-style-type: none"> • If tVSEO is 0 then VSEOFFSET = 0, else VSEOFFSET = HTOTAL - tVSEO 	16'b0
DPC Vertical Sync START Offset Register (DPCVSSOFFSET)				
Address : C010_2938h / C010_2D38h				

Bit	R/W	Symbol	Description	Reset Value
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	VSSOFFSET	<p>Specifies the number of clocks from start of the horizontal sync to start of the vertical sync, where the horizontal sync is last one in the vertical front porch. If this value is 0 then start of the vertical sync synchronizes with start of the horizontal sync which is new one in the vertical sync. This value has to be less than HTOTAL. When interlace mode, this value is used for odd field.</p> <ul style="list-style-type: none"> If tVSSO is 0 then VSSOFFSET = 0, else VSSOFFSET = HTOTAL - tVSSO 	16'b0
DPC Even Field Vertical Sync End Offset Register (DPCEVSEOFFSET)				
Address : C010_293Ch / C010_2D3Ch				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	EVSEOFFSET	<p>Specifies the number of clocks from start of the horizontal sync to end of the vertical sync, where the horizontal sync is last one in the vertical sync. If this value is 0 then end of the vertical sync synchronizes with start of the horizontal sync which is new one in the vertical back porch. This value has to be less than HTOTAL and is used for even field.</p> <ul style="list-style-type: none"> If tEVSEO is 0 then EVSEOFFSET = 0, else EVSEOFFSET = HTOTAL - tEVSEO 	16'b0
DPC Even Field Vertical Sync START Offset Register (DPCEVSSOFFSET)				
Address : C010_2940h / C010_2D40h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	EVSSOFFSET	<p>Specifies the number of clocks from start of the horizontal sync to start of the vertical sync, where the horizontal sync is last one in the vertical front porch. If this value is 0 then start of the vertical sync synchronizes with start of the horizontal sync which is new one in the vertical sync. This value has to be less than HTOTAL and is used for even field.</p> <ul style="list-style-type: none"> If tEVSSO is 0 then EVSSOFFSET = 0, else EVSSOFFSET = HTOTAL - tEVSSO 	16'b0
DPC Sync Delay 0 Register (DPCDELAY0)				
Address : C010_2944h / C010_2D44h				
[31:12]	R	RESERVED	Reserved	20'b0
[15]	R/W	PADPOLFIELD	Specifies the polarity of the PAD's field signal. 0 : Normal(Low is odd field) 1 : Inversion(Low is even field)	1b0
[14]	R	RESERVED	Reserved	1b0
[13:8]	R/W	DELAYVS	Specifies delay value of the vertical sync/FRM output. This value depends on the output format. The unit is VCLK2.	4'b0
[5:0]	R/W	DELAYHS	Specifies delay value of the horizontal sync/CP1 output. This value depends on the output format. The unit is VCLK2.	4'b0
DPC Sync UPScale control register 0(DPUPSCALECON0)				
Address : Not available / C010_2D48h				
[31:16]	R	RESERVED	Reserved	16'b0
[15:8]	R/W	UPSCALEL	<p>Specifies the ratio for the horizontal scale. The formula to calculate this value is as follows: When FILTERENB is 1 and the destination width is wider than the source width:</p> <ul style="list-style-type: none"> UPSCALE = (source width-1) * (1<<11) / (destination width-1) UPSCALEL = UPSCALE[7:0] 	8'b0
[7:1]	R	RESERVED	Reserved	7b0
[0]	R/W	UPSCALERENB	Decide whether to enlarge Source Image horizontal width. 0 : Scaler Disable 1 : Scaler Enable	1b0
DPC Sync UPScale control register 1(DPUPSCALECON1)				
Address : Not available / C010_2D4Ch				
[31:15]	R	RESERVED	Reserved	17b0
[14:0]	R/W	UPSCALEH	Specifies the ratio for the horizontal scale. The formula to calculate this value is as follows: When FILTERENB is 1 and the destination width is wider than the source width:	15'b0

Bit	R/W	Symbol	Description	Reset Value
			• UPSCALE = (MLCScreenwidth-1) * (1<<11) / (destination width-1) UPSCALEH = UPSCALE[22:8]	
Reserved				
Address : C010_294Ch ~2977h / C010_2D4Ch ~2D77h				
DPC Sync DELAY CONTROL register 1 (DPCDELAY1)				
Address : C010_2978h / C010_2D78h				
[31:6]	R	RESERVED	Reserved	26'b0
[5:0]	R/W	DELAYDE	Specifies delay value of DE(data enable)/CP2 output. This value depends on the output format. The unit is VCLK2.	6b0
DPC MPU i80 timing CONTROL register 0 (DPCmputime0)				
Address : C010_297Ch / C010_2D7Ch				
[31:16]	R	RESERVED	Reserved	16'b0
[15:8]	R/W	I80HOLDTIME	Specifies value of MPU i80 hold time config. The unit is PCLK.	8b0
[7:0]	R/W	I80SETUPTIME	Specifies value of MPU i80 setup timeconfig. The unit is PCLK.	8b0
DPC MPU i80 timing CONTROL register 1 (DPCmputime1)				
Address : C010_2980h / Not available				
[31:8]	R	RESERVED	Reserved	24'b0
[7:0]	R/W	I80ACCTIME	Specifies value of MPU i80 hold time config. The unit is PCLK.	8b0
DPC MPU i80 lowbit Write data register(dpcmpuwrdata)				
Address : C010_2984h / Not available				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	W	MPUWRDATA	Write low 16bit write data for MPU i80. Users must write this register first for MPU i80 access via CPU.	
DPC MPU i80 index Write data register(dpcmpuindex)				
Address : C010_2988h / Not available				
[31:8]	R	RESERVED	Reserved	16'b0
[7:0]	W	MPUINDEX	Write high 8bit write data for MPU i80 index access. (RS = 0)	
DPC MPU i80 status read data register(dpcmpustatus)				
Address : C010_298Ch / Not available				
[31:8]	R	RESERVED	Reserved	16'b0
[7:0]	R	MPUSTATUS	Read high 8bit write data1 for MPU i80 index access. (RS = 0)	
DPC MPU i80 READ/WRITE data register(dpcmpUDatah)				
Address : C010_2990h / Not available				
[31:8]	R	RESERVED	Reserved	16'b0
[7:0]	R/W	MPUDATAH	Read/Write high 8bit write data1 for MPU i80 data access. (RS = 1)	
DPC MPU i80 lowbit read data register(dpcmpurdata)				
Address : C010_2994h / Not available				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R	MPUWRDATA	Read low 16bit write data for MPU i80. Users must read this register after reading DPCMPUSTATUS or DPCMPUDATAH for MPU i80 access via CPU.	
Reserved				

Bit	R/W	Symbol	Description	Reset Value
Address : C010_2998h ~299Bh / C010_2D98h ~2D9Bh				
DPC MPU i80 command buffer lowbit data register(dpccmdbufferrdataL)				
Address : C010_299Ch / Not available				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	W	MPUCMDBUFL	Write low 16bit write data for MPU i80. Users must write this register first for MPU i80 command buffer	
DPC MPU i80 command buffer highbit data register(dpccmdbufferrdataH)				
Address : C010_29A0h / Not available				
[31:8]	R	RESERVED	Reserved	24'b0
[7:0]	W	MPUCMDBUFH	Write high 8bit write data for MPU i80.	
DPC MPU i80 timing CONTROL register 1 (DPCmpitime1)				
Address : C010_29A4h / C010_2DA4h				
[31:6]	R	RESERVED	Reserved	26'b0
[5]	R/W	POLDE	Specifies the polarity of the Data Enable output. 0 : Normal(High is data enable) 1 : Inversion(High is data enable)	1'b0
[4]	R/W	POLNCS	Specifies the polarity of the <i>nCS</i> signal. (MPU i80 type, primary only) 0 : Normal(Low is chip select) 1 : Inversion (High is chip select)	1'b0
[3]	R/W	POLNWR	Specifies the polarity of the <i>nWR</i> signal. (MPU i80 type, primary only) 0 : Normal(negedge trigger) 1 : Inversion(posedge trigger)	1'b0
[2]	R/W	POLNRD	Specifies the polarity of the <i>nRD</i> signal. (MPU i80 type, primary only) 0 : Normal(negedge trigger) 1 : Inversion(posedge trigger)	1'b0
[1]	R/W	POLRS	Specifies the polarity of the <i>RS</i> signal. (MPU i80 type, primary only) 0 : Normal(Low is index access) 1 : Inversion (High is index access)	1'b0
[0]	R/W	RESERVVED	Must be Set 1'b0	1'b0
DPC PAD LOCATION CONTROL register 0 (DPCpadposition0)				
Address : C010_29A8h / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC2	Select a location to go in the video data number 2 output. (active when i80 mode)	5b 0
[9:5]	R/W	PADLOC1	Select a location to go in the video data number 1 output. (active when i80 mode)	5b 0
[4:0]	R/W	PADLOC0	Select a location to go in the video data number 0 output. (active when i80 mode)	5b 0
DPC PAD LOCATION CONTROL register 1 (DPCpadposition1)				
Address : C010_29ACh / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC5	Select a location to go in the video data number 5 output. (active when i80 mode)	5b 0
[9:5]	R/W	PADLOC4	Select a location to go in the video data number 4 output. (active when i80 mode)	5b 0
[4:0]	R/W	PADLOC3	Select a location to go in the video data number 3 output. (active when i80 mode)	5b 0
DPC PAD LOCATION CONTROL register 2 (DPCpadposition1)				
Address : C010_29B0h / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC8	Select a location to go in the video data number 8 output. (active when i80 mode)	5b 0
[9:5]	R/W	PADLOC7	Select a location to go in the video data number 7 output. (active when i80 mode)	5b 0
[4:0]	R/W	PADLOC6	Select a location to go in the video data number 6 output. (active when i80 mode)	5b 0

Bit	R/W	Symbol	Description	Reset Value
DPC PAD LOCATION CONTROL register 2 (DPCpadposition1)				
Address :C010_29B4h / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC11	Select a location to go in the video data number 11 output. (active when i80 mode)	5'b0
[9:5]	R/W	PADLOC10	Select a location to go in the video data number 10 output. (active when i80 mode)	5'b0
[4:0]	R/W	PADLOC9	Select a location to go in the video data number 9 output. (active when i80 mode)	5'b0
DPC PAD LOCATION CONTROL register 2 (DPCpadposition1)				
Address :C010_29B8h / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC14	Select a location to go in the video data number 14 output. (active when i80 mode)	5'b0
[9:5]	R/W	PADLOC13	Select a location to go in the video data number 13 output. (active when i80 mode)	5'b0
[4:0]	R/W	PADLOC12	Select a location to go in the video data number 12 output. (active when i80 mode)	5'b0
DPC PAD LOCATION CONTROL register 2 (DPCpadposition1)				
Address :C010_29BCh / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC17	Select a location to go in the video data number 17 output. (active when i80 mode)	5'b0
[9:5]	R/W	PADLOC16	Select a location to go in the video data number 16 output. (active when i80 mode)	5'b0
[4:0]	R/W	PADLOC15	Select a location to go in the video data number 15 output. (active when i80 mode)	5'b0
DPC PAD LOCATION CONTROL register 2 (DPCpadposition1)				
Address :C010_29C0h / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC20	Select a location to go in the video data number 20 output. (active when i80 mode)	5'b0
[9:5]	R/W	PADLOC19	Select a location to go in the video data number 19 output. (active when i80 mode)	5'b0
[4:0]	R/W	PADLOC18	Select a location to go in the video data number 18 output. (active when i80 mode)	5'b0
DPC PAD LOCATION CONTROL register 2 (DPCpadposition1)				
Address :C010_29C4h / Not available				
[31:15]	R	RESERVED	Reserved	17'b0
[14:10]	R/W	PADLOC23	Select a location to go in the video data number 23 output. (active when i80 mode)	5'b0
[9:5]	R/W	PADLOC22	Select a location to go in the video data number 22 output. (active when i80 mode)	5'b0
[4:0]	R/W	PADLOC21	Select a location to go in the video data number 21 output. (active when i80 mode)	5'b0
DPC PAD LOCATION MASK register 0 (DPCRGBMASK0)				
Address :C010_29C8h / Not available				
[31:16]	R	RESERVED	Reserved	16'b0
[15:0]	R/W	RGBMASK[15:0]	Specifies the mask of the video data output (active when i80 mode) 0:Masked (output = 0)1 : Enabled	16'b0
DPC PAD LOCATION MASK register 1 (DPCRGBMASK1)				
Address :C010_29CCh / Not available				
[31:8]	R	RESERVED	Reserved	24'b0
[7:0]	R/W	RGBMASK[23:16]	Specifies the mask of the video data output (active when i80 mode) 0:Masked (output = 0)1 : Enabled	8'b0

Bit	R/W	Symbol	Description	Reset Value
DPC RGB SHIFT CONTROL register(DPCRGBSHIFT)				
Address :C010_29D0h / Not available				
[31:6]	R	RESERVED	Reserved	26'b0
[5]	R	FIELDFLAG	Specifies the field flag of the interface mode 0 : odd frame 1 : even frame	
[4:0]	R/W	RGBSHIFT	Specifies the shift number of the video data output (active when i80 mode) PAD video data[23:0] = video data[23:0]<<RGBSHIFT	5'b0
DPC MPU i80 command buffer FLUSH Control register(dpcdataflush)				
Address :C010_29D4h / Not available				
[31:4]	R	RESERVED	Reserved	28'b0
[4]	W	REGFLUSH	Register flush manualy (primary only)	
[3]	R	CMDBUFFULL	MPU i80 command buffer's full flag 0 : Not full 1 : Full	1b0
[2]	R	CMDBUFEMPTY	MPU i80 command buffer's empty flag 0 : Not empty 1 : Empty	1b0
[1]	R/W	CMDBUFFLUSH	Set the dirty flag for MPU i80 command buffer 1: when vertical sync period, command buffer start to send data.	1b0
[0]	W	CMDBUFCCLR	Clear the MPU i80 command buffer	
Reserved				
Address : C010_29D8h ~2BCFh / C010_2DD8h ~2FCFh				
DPC Clock Generation Enable Register (DPCCLKENB)				
Address : C010_2BC0h / C010_2FC0h				
[31:4]	R	RESERVED	Reserved	28'b0
[3]	R/W	PCLKMODE	Specifies PCLK operating mode. 0 : PCLK is only enabled when CPU accesses this module 1 : PCLK is always enabled	1b0
[2]	R/W	CLKGENENB	Enable/Disable to generate a clock. 0 : Disable 1 : Enable	1b0
[1:0]	R	RESERVED	Reserved	2b0
DPC Clock Generation Control 0 Low Register (DPCCLKGEN0L)				
Address : C010_2BC4h / C010_2FC4h				
[31:16]	R	RESERVED	Reserved	16b0
[15]	R/W	OUTCLKENB	Specifies the direction of the PADVCLK pad. This bit should be set when CLKSRCSEL0/1 is 3 or 4. 0 : Enable(Output) 1 : Reserved	1b0
[14:13]	R	RESERVED	Reserved	2b0
[12:5]	R/W	CLKDIV0	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL0	Specifies the source clock. 0 : PLL0 1 : PLL1 2 : SVLCK 3 : P(S)VLC 4 : ~P(S)VCLK 5 : AVCLK 6 : ~SVLCK 7 : Reserved	3b0
[1]	R/W	OUTCLKINV0	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	RESERVED	Reserved (This bit should be set to '0')	1b0

Bit	R/W	Symbol	Description	Reset Value
DPC Clock Generation Control 0high Register (DPCCLKGEN0h)				
Address : C010_2BC8h/C010_2FC8h				
[31:5]	R/W	RESERVED	Reserved for future use. This bit should be set to '0'	27b0
[4:0]	R/W	OUTCLKDELAY0	Specifies delay value of the clock output. This value is needed to adjust clock skew with respect to data signal. TBD	5b0
DPC Clock Generation Control 1 low Register (DPCCLKGEN1L)				
Address : C010_2BCCh/C010_2FCCh				
[31:13]	R	RESERVED	Reserved	19b0
[12:5]	R/W	CLKDIV1	Specifies divider value of the source clock. Divider value = CLKDIV0 + 1	8b0
[4:2]	R/W	CLKSRCSEL1	Specifies the source clock. 0 : PLL0 1 : PLL1 2 : SVLCK 3 : P(S)VLCK 4 : ~P(S)VLCK 5 : AVCLK 6 : ~SVLCK 7 : Reserved	3b0
[1]	R/W	OUTCLKINV1	Specifies whether to invert the clock output. 0 : Normal (Falling Edge) 1 : Invert (Rising Edge)	1b0
[0]	R/W	OUTCLKSEL	1/2 ns delay is applied to Out clock from source clock Can be used only in case of division by even number 0 : Bypass 1 : source clock /2 ns	1b0
DPC Clock Generation Control 1high Register (DPCCLKGEN1h)				
Address : C010_2BD0h/C010_2FD0h				
[31:5]	R/W	RESERVED	Reserved for future use. You don't have to write any value except 0.	27b0
[4:0]	R/W	OUTCLKDELAY1	Specifies delay value of the clock output. This value is needed to adjust clock skew with respect to data signal. TBD	5b0

Section 35. Scaler

35.1 Overview

The Scaler is the block to change image sizes. The Scaler reads an image from the memory and writes the image to the memory after Up/Down Scaling and Low-pass Filtering. At this time, the Scaler changes the direction of the image by using the Flip or Rotation function.

35.1.1 Features

- Source/Destination Image
- Format: Separated YUV Format(420, 422, 444), Interleaved UV
- Size: (8~4096) x (8~4096) (Width is set as a multiple of eight).
- Upscale Ratio: 8x8 -> 4096x4096
- Downscale Ratio: 4096x4096 -> 8x8
- Lowpass Filter available after Upscale or before Downscale.
- Horizontal 5-Tab Filter: Coefficients 64 Sets.
- Vertical 3-Tab Filter: Coefficients 32 Sets (For Frequency Response, refer to Operation Item).

35.1.2 Block Diagram

The Scaler consists of the blocks (SRC_ADDR_GEN, DEST_ADDR_GEN) to generate addresses, the Filter block, the FIFO block, and the blocks (CPUIF and POS_GEN2) to exchange data with bus.

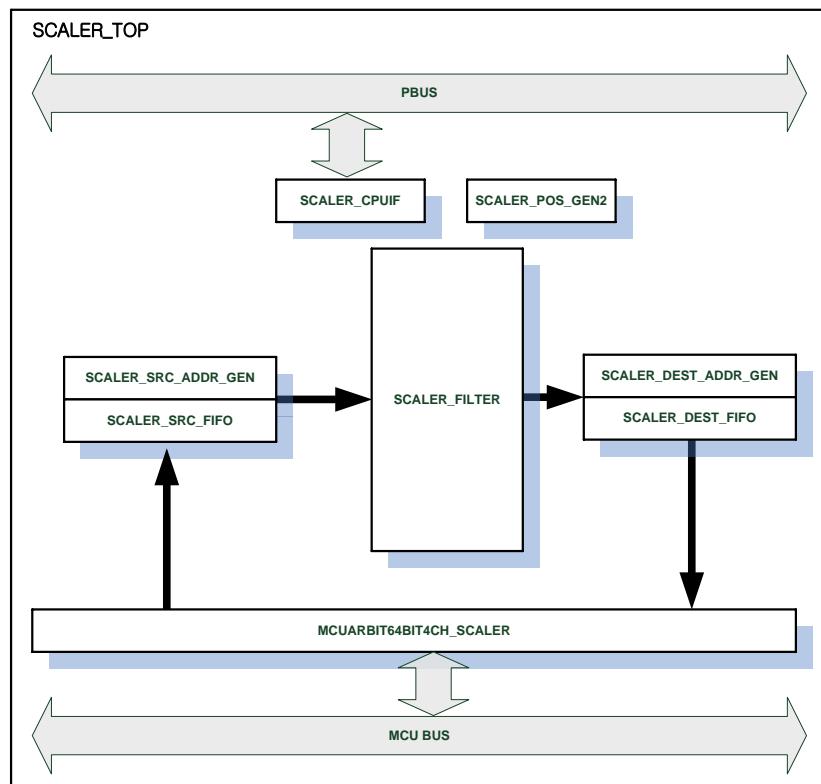


Figure 35-1 FINE SCALER FILTER Block Diagram

35.2 Functional Description

The Scales enable a user to read a source image in memory, change the image size and store the image in thememory. The Scaler changes image sizes by using the setting values of the address, width and height of thesource and destination images. A user can use the low-pass filter and rotate functions of the Scaler.

35.2.1 DIGITAL FILTER CHARACTERISTICS

The low-pass filter of the Scaler has the horizontal filter of 5-tab and the vertical filter of 3-tab. The Scaler preventsimage quality deterioration when an image is enlarged by using the low-pass filter.

35.2.1.1 Horizontal Filter (5-Tab FIR Filter) Frequency Response and Group Delay

Figure 35-2 shows the characteristics of the horizontal filter of the Scaler and the filter has the setting rangebetween 0 and 63. The Scaler register, SCCFGREG.SC_HFILT_COEFF, is used for the setting.

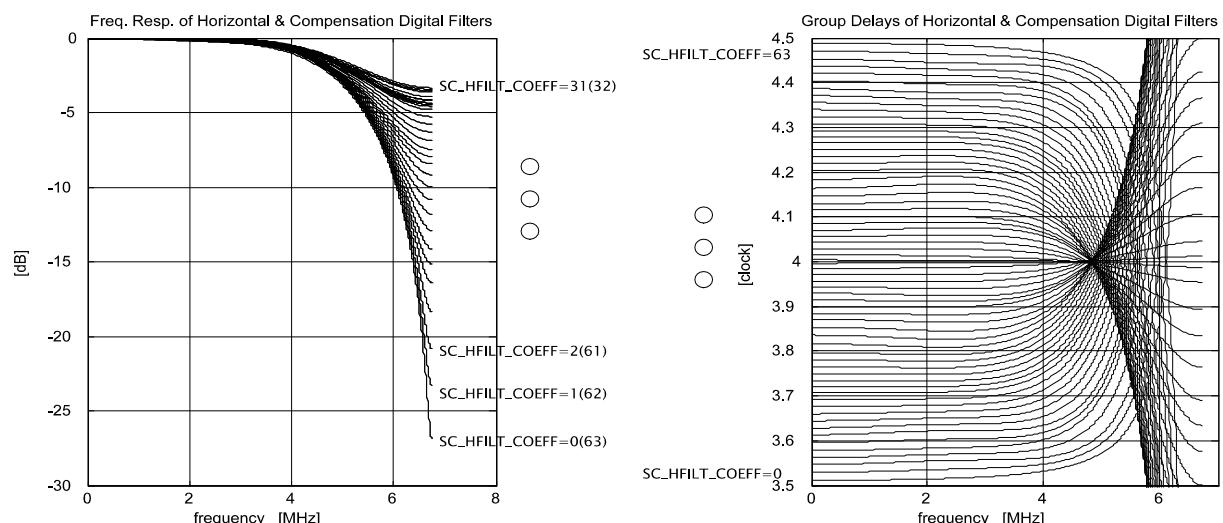


Figure 35-2 Horizontal Filter (5-Tab FIR Filter) Frequency Response and Group Delay

35.2.1.2 Vertical Filter (3-Tab FIR Filter) Frequency Response and Group Delay

Figure 35-3 shows the characteristics of the vertical filter of Scaler and the filter has the setting range between 0and 31. The Scaler register, SCCFGREG.SC_VFILT_COEFF, is used for the setting.

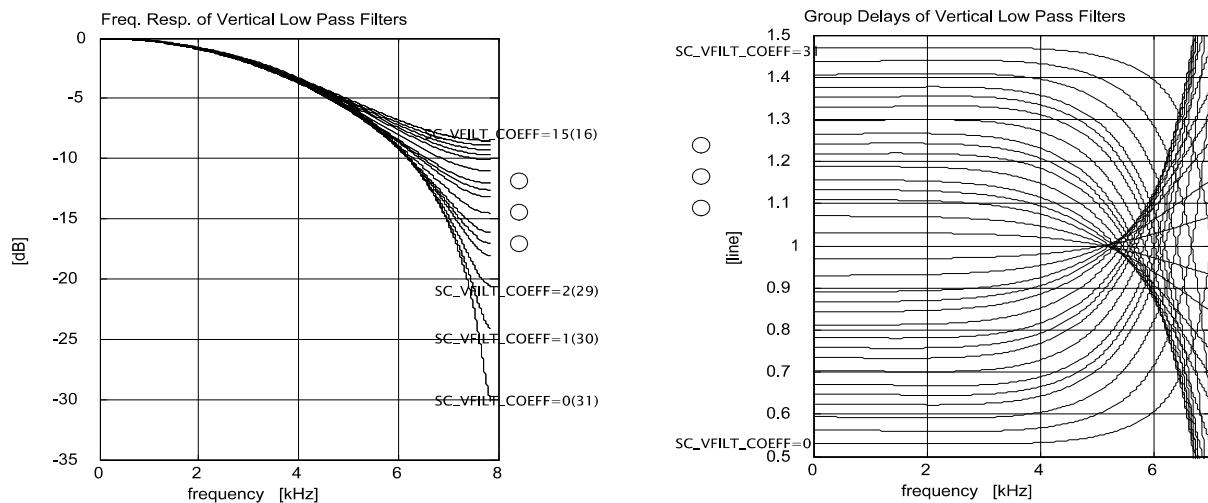


Figure 35-3 Vertical Filter (3-Tap FIR Filter) Frequency Response and Group Delay

35.3 Programming Guide

35.3.1 Configuration

- 1) Check the Scaler controller status : SCINTREG.SC_BUSY = 0
- 2) Set the source address and stride to the SCSRCADDR and SCSRCSTRIDE register respectively.
- 3) Set SCSRCSIZEREG.SC_SRC_WIDTH and SCSRCSIZEREG.SC_SRC_HEIGHT register. (the width is a multiple of 8).
- 4) Set the destination address and stride to the SCDESTADDR0 and SCDESTSTRIDE0 register respectively. If you need to scale UV interleaved image, set the destination address and stride to the SCDESTADDR1 and SCDESTSTRIDE1 register respectively.
- 5) Set SCDESTSIZEREG.SC_DEST_WIDTH and SCDESTSIZEREG.SC_DEST_HEIGHT register.(the width is a multiple of 8).
- 6) Delta Image setting : Set DELTAXREG and DELTAYREG register.
- 7) Soft setting : Set Horizontal/Vertical Set HVSOFTREG register.
- 8) Set the filter: Set the filter as On/Off by using SCCFGREG.SC_FILT_ENB. Select the Filter Coefficient Set byusing SCCFGREG.SC_HFILT_COEFF and SCCFGREG.SC_VFILT_COEFF.
- 9) Set the interrupt: SCINTREG.SC_INT_ENB register.

35.3.2 RUN

- 1) Set the SCRUNREG.SC_RUN bit as '1'.
- 2) Read the SCINTREG.SC_BUSY register to check the operation status.
- 3) If the SCRUNREG.SC_RUN bit is cleared when SCINTREG.SC_BUSY = 1, the operation halts immediately.

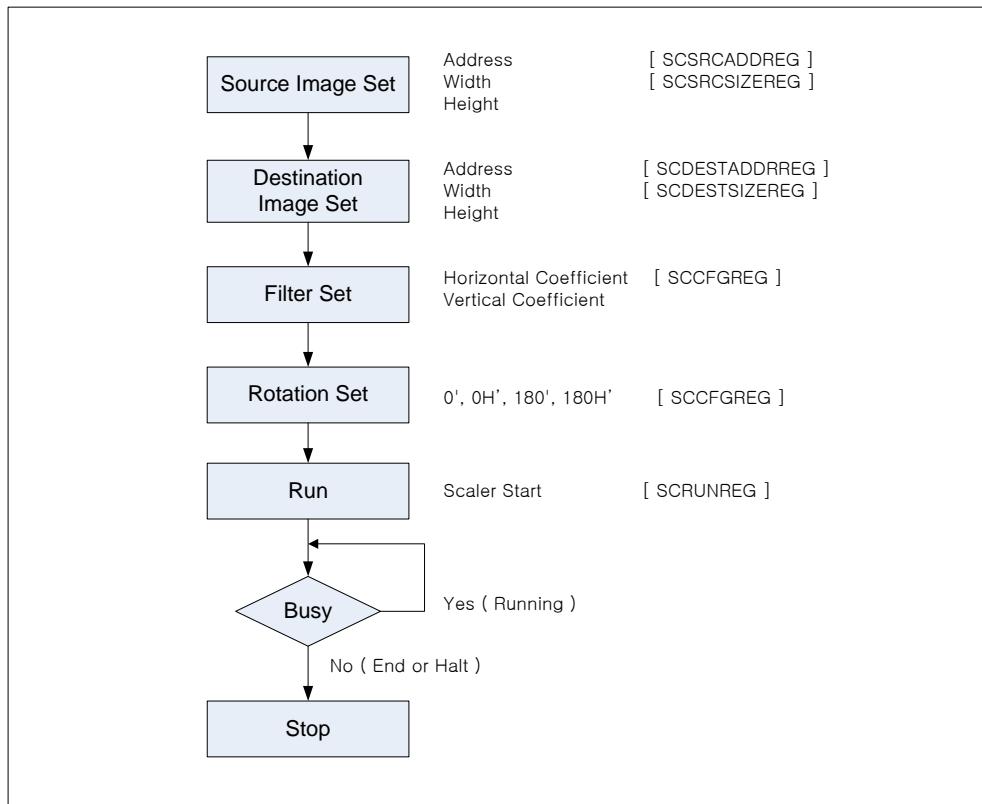


Figure 35-4 Scaler Operation Flow

35.4 Register Summary

Bit	R/W	Symbol	Description	Reset Value
Scaler Run Register (SCRUNREG)				
<i>Address : C006_6000h</i>				
[31:1]	-	RESERVED	Reserved	31'b0
[0]	R/W	SC_RUN	<p>Scaler RUN bit. When the scaling process is finished, SCALER clears this bit automatically. While the scale process is running, it can be halted by clearing this bit.</p> <p>0: Stop 1: Start (Auto Clear)</p>	1'b0
Scaler Configuration Register (SCCFGREG)				
<i>Address : C006_6004h</i>				
[31:21]	-	RESERVED	Reserved	11'b0
[20:16]	R/W	SC_VFILT_COEFF	<p>Vertical filter coefficient select. Range is 0 ~ 31. (See Frequency Response Graph)</p>	5'bx
[15:14]	-	RESERVED	Reserved	2'b0
[13:8]	R/W	SC_HFILT_COEFF	<p>Horizontal filter coefficient select. Range is 0 ~ 63. (See Frequency Response Graph)</p>	6'bx
[7:2]	-	RESERVED	Reserved	6'b0
[1:0]	R/W	SC_FILT_ENB	<p>Fine scale filter enable. 00: Filter Disable 01: Reserved 10: Reserved 11: Filter Enable</p> <p>Note : This bit should be set as '00' or '11'</p>	2'bx
Scaler Interrupt Register (SCINTREG)				
<i>Address : C006_6008h</i>				
[31:25]	-	RESERVED	Reserved	7'b0
[24]	R	SC_BUSY	<p>Scaler Busy Check 0: I2C-bus not busy (when read), I2C-bus interface STOP signal generation (when write) 1: I2C-bus busy (when read), I2C-bus interface START signal generation (when write)</p>	1'b0
[23:18]	-	RESERVED	Reserved	6'b0
[17]	R/W	CMD_PROC_INT_ENB	<p>Internal Command Processor Interrupt Enable 0: Disable 1: Enable</p>	1'b0
[16]	R/W	SC_INT_ENB	<p>Scaler Interrupt Enable 0: Disable 1: Enable</p>	1'b0
[15:10]	R/W	RESERVED	Reserved	6'b0
[9]	W	CMD_PROC_INT_CLR	<p>Clear Internal Command Processor Interrupt Pending Bit 0: None 1: Clear Interrupt Pending</p>	1'b0
[8]	W	SC_INT_CLR	<p>Clear Internal Command Processor Interrupt Pending Bit 0: None 1: Clear Interrupt Pending</p>	1'b0
[7:2]	-	RESERVED	Reserved	6'b0
[1]	R	CMD_PROC_INT_PEND	Command Processor Interrupt Pending Bit	1'b0

Bit	R/W	Symbol	Description	Reset Value
			0: None 1: Interrupt Pending	
[0]	R	SC_INT_PEND	Scaler Interrupt Pending Bit 0: None 1: Interrupt Pending	1'b0
Scaler Source Address Register (SCSRCADDRREG)				
<i>Address : C006_600Ch</i>				
[31:0]	R/W	SC_SRC_REG	Source Base Address Register	32'bx
Scaler Source Stride Register (SCSRCADDRREG)				
<i>Address : C006_6010h</i>				
[31:0]	R/W	SC_SRC_STR	Source Stride Register	32'bx
Scaler Source Size Register (SCSRCSIZEREG)				
<i>Address : C006_6014h</i>				
[31:28]	-	RESERVED	Reserved	4'bx
[27:16]	R/W	SC_SRC_HEIGHT	Source Image Height. Height's range is 8~4096 (Source Height -1)	12'b0
[15:12]	-	RESERVED	Reserved	4'b0
[11:0]	R/W	SC_SRC_WIDTH	Set Source Image Width. Width's range is 8~4096. Width must align to 8 (Source width -1)	12'bx
Scaler Destination Address Register 0 (SCDESTADDR0)				
<i>Address : C006_6018h</i>				
[31:0]	R/W	SC_DEST_ADDR0	Destination Base Address0	32'bx
Scaler Destination Stride Register 0 (SCDESTSTREDE0)				
<i>Address : C006_601Ch</i>				
[31:0]	R/W	SC_DEST_STRIE0	Destination Stride	32'bx
Scaler Destination Address Register1 (SCDESTADDR1)				
<i>Address : C006_6020h</i>				
[31:0]	R/W	SC_DEST_ADDR1	Destination Base Address1 Note : UV interleaved mode only	32'bx
Scaler Destination Stride Register1 (SCDESTADDR1)				
<i>Address : C006_6024h</i>				
[31:0]	R/W	SC_DEST_STRIE1	Destination Stride Note : UV interleaved mode only	32'bx
Scaler Destination Size Register (SCDESTSIZE)				
<i>Address : C006_6028h</i>				
[31:28]	-	RESERVED	Reserved	4'b0
[27:16]	R/W	SC_DEST_HEIGHT	Destination Image Height Height's range is 8 ~ 4096 (Destination Height -1)	12'bx
[15:12]	-	RESERVED	Reserved	4'b0
[11:0]	R/W	SC_DEST_WIDTH	Destination Image Width Image's range is 8 ~ 4096	12'bx
Scaler Horizontal Delta Register (DELTA_XREG)				

Bit	R/W	Symbol	Description	Reset Value
Address : C006_602Ch				
[31:0]	R/W	DELTAXREG	Delta X of X-axis $\text{DELTA_X} = (\text{sc_src_width} * \text{h'10000}) / (\text{sc_dest_width}-1)$	32'b0
Scaler Vertical Delta Register (DELTAYREG)				
Address : C006_6030h				
[31:0]	R/W	DELTAYREG	Delta Y of Y-axis $\text{DELTA_Y} = (\text{sc_src_height} * \text{h'10000}) / (\text{sc_dest_height}-1)$	32'b0
Scaler Ratio Reset Value Register (HVSOFTREG)				
Address : C006_6034h				
[31:21]	-	RESERVED	Reserved	11'bX
[20:16]	R/W	V_RATIO	Vertical Filter Ratio	5'b0
[15:6]	-	RESERVED	Reserved	10'bX
[5:0]	R/W	H_RATIO	Horizontal Filter Ratio	6'b0
Scaler Command Buffer Base Address Register (CMDBUFADDR)				
Address : C006_6038h				
[31:0]	R/W	CMDBUFADDR	Scaler Command Buffer Base Address Register	32'bX
Scaler Command Buffer Control Register (CMDBUFCON)				
Address : C006_603Ch				
[31:2]	-	RESERVED	Reserved	30'bX
[1]	R/W	CMDBUF_STOP	Scaler Command Buffer Stop Register 0: No operation 1: Stop	1'b0
[0]	R/W	CMDBUF_START	Scaler Command Buffer Start Register 0: No operation 1: Start	1'b0
Scaler YV Filter[N] Value Table Register0 (YVFILTER[N]_00_03)				
Address : Filter1 C006_6040h, Filter2 C006_6060h, Filter3 C006_6080				
[31:24]	W	FILTER_YVCOEF[N]_03	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N]_02	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N]_01	Scaler YVCOEF[N] value	8'b0
[7:0]	W	FILTER_YVCOEF[N]_00	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter[N] Value Table Register1 (YVFILTER[N]_04_07)				
Address : Filter1 C006_6044h, Filter2 C006_6064h, Filter3 C006_6084				
[31:24]	W	FILTER_YVCOEF[N]_07	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N]_06	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N]_05	Scaler YVCOEF[N] value	8'b0
[7:0]	W	FILTER_YVCOEF[N]_04	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter[N] Value Table Register2 (YVFILTER[N]_08_11)				
Address : Filter1 C006_6048h, Filter2 C006_6068h, Filter3 C006_6088				
[31:24]	W	FILTER_YVCOEF[N]_11	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N]_10	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N]_09	Scaler YVCOEF[N] value	8'b0

Bit	R/W	Symbol	Description	Reset Value
[7:0]	W	FILTER_YVCOEF[N_08]	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter[N] Value Table Register3 (YVFILTER[N_12_15])				
<i>Address : Filter1 C006_604Ch, Filter2 C006_606Ch, Filter3 C006_608C</i>				
[31:24]	W	FILTER_YVCOEF[N_15]	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N_14]	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N_13]	Scaler YVCOEF[N] value	8'b0
[7:0]	W	FILTER_YVCOEF[N_12]	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter[N] Value Table Register4 (YVFILTER[N_16_19])				
<i>Address : Filter1 C006_6050h, Filter2 C006_6070h, Filter3 C006_6090h</i>				
[31:24]	W	FILTER_YVCOEF[N_19]	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N_18]	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N_17]	Scaler YVCOEF[N] value	8'b0
[7:0]	W	FILTER_YVCOEF[N_16]	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter[N] Value Table Register5 (YVFILTER[N_20_23])				
<i>Address : Filter1 C006_6054h, Filter2 C006_6074h, Filter3 C006_6094h</i>				
[31:24]	W	FILTER_YVCOEF[N_23]	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N_22]	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N_21]	Scaler YVCOEF[N] value	8'b0
[7:0]	W	FILTER_YVCOEF[N_20]	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter[N] Value Table Register6 (YVFILTER[N_24_27])				
<i>Address : Filter1 C006_6058h, Filter2 C006_6078h, Filter3 C006_6098h</i>				
[31:24]	W	FILTER_YVCOEF[N_27]	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N_26]	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N_25]	Scaler YVCOEF[N] value	8'b0
[7:0]	W	FILTER_YVCOEF[N_24]	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter[N] Value Table Register7 (YVFILTER[N_28_31])				
<i>Address : Filter1 C006_605Ch, Filter2 C006_607Ch, Filter3 C006_609Ch</i>				
[31:24]	W	FILTER_YVCOEF[N_31]	Scaler YVCOEF[N] value	8'b0
[23:16]	W	FILTER_YVCOEF[N_30]	Scaler YVCOEF[N] value	8'b0
[15:8]	W	FILTER_YVCOEF[N_29]	Scaler YVCOEF[N] value	8'b0
[7:0]	W	FILTER_YVCOEF[N_28]	Scaler YVCOEF[N] value	8'b0
Scaler YV Filter2 Value Table Register0 (YVFILTER2_00_03)				
<i>Address : C006_6060h</i>				
[31:24]	W	FILTER_YVCOEF2_03	Scaler YVCOEF2 value	8'b0
[23:16]	W	FILTER_YVCOEF2_02	Scaler YVCOEF2 value	8'b0
[15:8]	W	FILTER_YVCOEF2_01	Scaler YVCOEF2 value	8'b0
[7:0]	W	FILTER_YVCOEF2_00	Scaler YVCOEF2 value	8'b0

Bit	R/W	Symbol	Description	Reset Value
Reserved				
Address : C006_60A0h ~ C006_60FC				
Scaler YH Filter1~5 Value Table Register0 (YHFILTER[N]_00_01)				
Address : Filter1 C006_6100h, Filter2 C006_6140, Filter3 C006_6180, Filter4 C006_61C0, Filter5 C006_6200				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_01	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_00	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register1 (YHFILTER[N]_02_03)				
Address : Filter1 C006_6104h, Filter2 C006_6144, Filter3 C006_6184, Filter4 C006_61C4, Filter5 C006_6204				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_03	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_02	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register2 (YHFILTER[N]_04_05)				
Address : Filter1 C006_6108h, Filter2 C006_6148, Filter3 C006_6188, Filter4 C006_61C8, Filter5 C006_6208				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_05	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_04	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register3 (YHFILTER[N]_06_07)				
Address : Filter1 C006_610Ch, Filter2 C006_614C, Filter3 C006_618C, Filter4 C006_61CC, Filter5 C006_620C				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_07	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_06	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register4 (YHFILTER[N]_08_09)				
Address : Filter1 C006_6110h, Filter2 C006_6150, Filter3 C006_6190, Filter4 C006_61D0, Filter5 C006_6210				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_09	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_08	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register5 (YHFILTER[N]_10_11)				
Address : Filter1 C006_6114h, Filter2 C006_6154, Filter3 C006_6194, Filter4 C006_61D4, Filter5 C006_6214				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_11	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_10	Scaler YHCOEF[N] value	10'b0

Bit	R/W	Symbol	Description	Reset Value
Scaler YH Filter1~5 Value Table Register6 (YHFILTER[N]_12_13)				
<i>Address :Filter1 C006_6118h, Filter2 C006_6158, Filter3 C006_6198, Filter4 C006_61D8, Filter5 C006_6218</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_13	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_12	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register7 (YHFILTER[N]_14_15)				
<i>Address :Filter1 C006_611Ch, Filter2 C006_615C, Filter3 C006_619C, Filter4 C006_61DC, Filter5 C006_621C</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_15	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_14	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register8 (YHFILTER[N]_16_17)				
<i>Address :Filter1 C006_6120h, Filter2 C006_6160, Filter3 C006_61A0, Filter4 C006_61E0, Filter5 C006_6220</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_17	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_16	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register9 (YHFILTER[N]_18_19)				
<i>Address :Filter1 C006_6124h, Filter2 C006_6164, Filter3 C006_61A4, Filter4 C006_61E4, Filter5 C006_6224</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_19	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_18	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register10 (YHFILTER[N]_20_21)				
<i>Address :Filter1 C006_6128h, Filter2 C006_6168, Filter3 C006_61A8, Filter4 C006_61E8, Filter5 C006_6228</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_21	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_20	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register11 (YHFILTER[N]_22_23)				
<i>Address :Filter1 C006_612Ch, Filter2 C006_616C, Filter3 C006_61AC, Filter4 C006_61EC, Filter5 C006_622C</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_23	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_22	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register12 (YHFILTER[N]_24_25)				
<i>Address :Filter1 C006_6130h, Filter2 C006_6170, Filter3 C006_61B0, Filter4 C006_61F0, Filter5 C006_6230</i>				

Bit	R/W	Symbol	Description	Reset Value
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_25	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_24	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register13 (YHFILTER[N]_26_27)				
<i>Address :Filter1 C006_6134h, Filter2 C006_6174, Filter3 C006_61B4, Filter4 C006_61F4, Filter5 C006_6234</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_27	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_26	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register13 (YHFILTER[N]_28_29)				
<i>Address :Filter1 C006_6138h, Filter2 C006_6178, Filter3 C006_61B8, Filter4 C006_61F8, Filter5 C006_6238</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_29	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_28	Scaler YHCOEF[N] value	10'b0
Scaler YH Filter1~5 Value Table Register15 (YHFILTER[N]_30_31)				
<i>Address :Filter1 C006_613Ch, Filter2 C006_617C, Filter3 C006_61BC, Filter4 C006_61FC, Filter5 C006_623C</i>				
[31:26]	-	RESERVED	Reserved	6'bx
[25:16]	W	FILTER_YHCOEF[N]_31	Scaler YHCOEF[N] value	10'b0
[15:10]	-	RESERVED	Reserved	6'x
[9:0]	W	FILTER_YHCOEF[N]_30	Scaler YHCOEF[N] value	10'b0

Section 36. LVDS

36.1 Overview

The LVDS(Low-Voltage differential signaling) is a block that generates the signals to interface with external LVDS display devices. The LVDS consists of a LVDS Controller and LVDS PHY block. The LVDS Controller receives RGB Video data from the DPC (or Resolution Converter) and converts RGB Video Data into a suitable LVDS data stream format and transmits converted RGB Video data to the LVDS PHY block. And the LVDS Controller transmits the control signals to the LVDS PHY block. The LVDS PHY block transmits received RGB Video Data from the LVDS Controller through 6 LVDS output channels.

36.1.1 Features

- Selectable Progressive RGB Video data (2 DPC)
- Supports JEIDA and VESA data packing for LVDS output
- Programmable data packing for LVDS output (configurable)
- Internal Input Video clock range : 30M to 90 MHz
- 6 LVDS output channels (5 data channels, 1 clock channel)
- 35:7 data channel compression up to 630Mbps on each LVDS channel
- Power down mode

36.1.2 Block Diagram

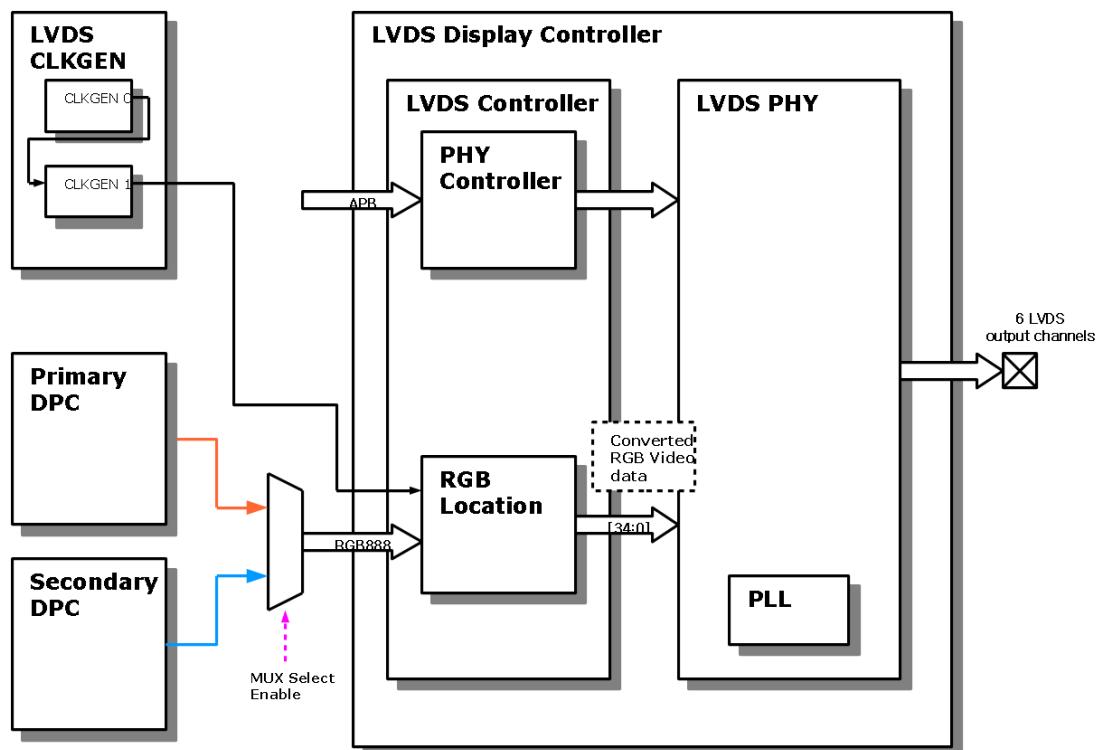


Figure 36-1. LVDS Block Diagram

36.2 Functional Description

36.2.1 LVDS data packing format

The LVDS Controller converts received RGB Video Data from DPC into one of three data packing formats and transmits them to the LVDS PHY block. Following is supported LVDS data packing formats.

- VESA data packing format
- JEIDA data packing format
- User Programmable data packing format

In the VESA and JEIDA data packing formats, four of the five data channel are used for transmission. Following is a format for each data packing format.

Channel	Bit Position						
	6	5	4	3	2	1	0
LVDS Channel A	bit 0	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LVDS Channel B	bit 1	bit 0	bit 5	bit 4	bit 3	bit 2	bit 1
LVDS Channel C	DE	VSYNC	Hsync	bit 5	bit 4	bit 3	bit 2
LVDS Channel D	Don't care	bit 7	bit 6	bit 7	bit 6	bit 7	bit 6
LVDS Channel E	Not used	Not used	Not used	Not used	Not used	Not used	Not used

Table 36-1. VESA data packing format

Channel	Bit Position						
	6	5	4	3	2	1	0
LVDS Channel A	bit 2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2
LVDS Channel B	bit 3	bit 2	bit 7	bit 6	bit 5	bit 4	bit 3
LVDS Channel C	DE	VSYNC	Hsync	bit 7	bit 6	bit 5	bit 4
LVDS Channel D	Don't care	bit 1	bit 0	bit 1	bit 0	bit 1	bit 0
LVDS Channel E	Not used	Not used	Not used	Not used	Not used	Not used	Not used

Table 36-2. JEIDA data packing format

When users set *LCD_FORMAT* bit of *LVDSCTRL0* register to VESA data packing format, the outputs from the LVDS has a form of Table 36-1. Similarly, when users set them to JEIDA data packing format, the outputs has a form of Table 36-2.

In the User Programmable data packing format, users can set own data packing format for the LVDS output. Users determine for each bit of the channel, willing to output what signal. The signal can be a Hsync, Vsync, DE, or RGB Color bit. In the User Programmable data packing format, the setting value of the register means in Table 36-3.

Bit Position							
[34:31]	[30:27]	[26]	[25]	[24]	[23:16]	[15:8]	[7:0]
4'b1111	4'b0000	VDEN	VSync	HSync	RED[7:0]	GREEN[7:0]	BLUE[7:0]

Table 36-3. Location Setting Input Format

For examples, if users want to set 0'st bit of the LVDS Channel A to RED[2], users set the *LOC_A0* bit of the *LVDSLOC0* register to 17. \$\$\$ (Note that 17'th bit position in Table 36-3 is the RED[2]) If users want to set 5'th bit of the LVDS Channel D to VSync, users set the *LOC_D5* bit of the *LVDSLOC5* register to 25.

36.2.2 LVDS Application Note

In NXP4330D/Q, users can choose one of two RGB Video data.

- Primary DPC Video Data
- Secondary DPC Video Data

The following sequence should be used to use LVDS device. (Assuming that the sequence uses the Primary DPC Video Data)

- 1) Release the Reset of the DisplayTop and DualDisplay
- 2) Set the configuration of the Primary MLC and the Primary DPC. Set the Vertical, Horizontal Sync width of the DPC with the same width of the exterior LVSD Display Device's. Set the output format of the DPC to RGB888 format. Assuming that the DPC CLKGEN's clock source is PLL2 and clock divisor is 2.
- 3) Set the LVDS CLKGEN's clock source is PLL2 and clock divisor is 2 (same as the DPC CLKGEN). Recommend Configuration is the following.
 - a) CLKGEN 0 : SRC(PLL2), DIV(2)
 - b) CLKGEN 1 : SRC(7), DIV(2)
- 4) Set the *LVDS_MUXSEL* bit of the *LVDS_MUXCTRL* register to 0 (Using the Primary DPC), Set the *LVDS_MUXENB* bit of the *LVDS_MUXCTRL* register to 1 (Enable).
- 5) Set the LVDS data packing mode, Set the configuration for LVDS Control Register. Table 36-4 shows a example of using VESA data packing format.

Register Name	Value
LVDSCTRL0	0x10036C70
LVDSCTRL1	0x000036DB
LVDSCTRL2	0x0000538E (High Speed, >90MHz) 0x0000128A (Low Speed, <90MHz)
LVDSCTRL3	0x00000333
LVDSCTRL4	0x03FFC20
LVDSTMODE0	0x00000080

Table 36-4. Recommand Config for LVDS (VESA)

- 6) Release the reset of the LVDS PHY block.

36.2.3 Skew control between output data and clock

The LVDS has the auto de-skew control function, If the *LVDSCTRL0.I.AUTO_SEL* and *LVDSCTRL4.AUTO_DSK_SEL* are high. The LVDS could operate the de-skewing function automatically. For the auto de-skewing, the signal which informs the vertical blank region is injected to the LVDS TX as like Figure 36-2.

If customer want to implement the auto de-skew function, RX have to support auto deskew function.

In NXP4330D/Q, auto de-skew works only with Using the Primary DPC RGB Video.

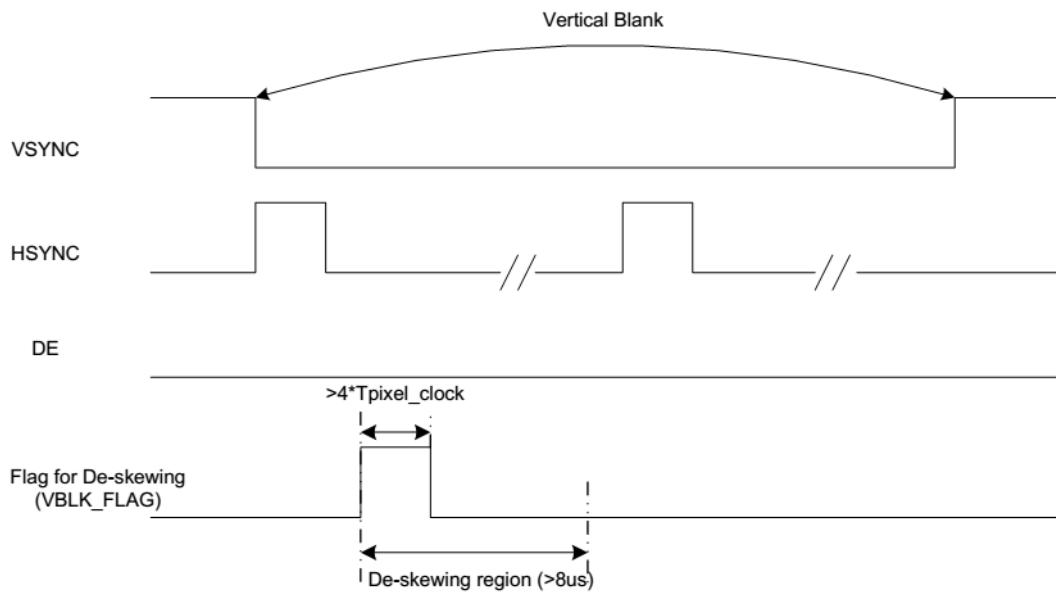


Figure 36-2. De-skewing timing diagram at Vertical blank

36.2.4 Electrical characteristics

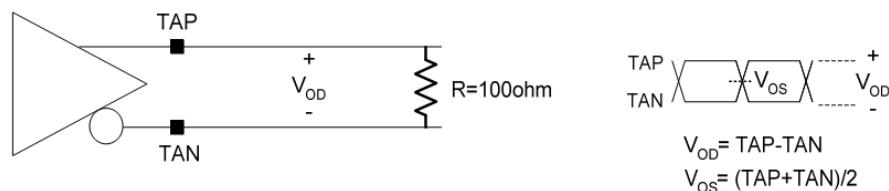


Figure 36-3. Output Common mode volatage and differential voltage

36.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
LVDS Control register 0 (LVDSCTRL0)				
<i>Address : C010A000h</i>				
[31]	R	RESERVED	Reserved	1'b0
[30]	R/W	CPU_I_VBLK_FLAG_SEL	Enable/Disable Using CPU_I_VBLK_FLAG (I_VBLK_FLAG used for de-skew) 0 : Using Input Video's VBLK 1 : Using CPU_I_VBLK	1'b0
[29]	R/W	CPU_I_VBLK_FLAG	Specifies the CPU_I_VBLK_FLAG Value	1'b0
[28]	R/W	SKINI_BST	Delay initial control pin for BIST 0 : bypass 1 : Delay control	1'b0
[27]	R/W	DLYS_BST	Delay control pin for BIST 0:25ps 1:50ps	1'b0
[26]	R/W	I_AUTO_SEL	Auto de-skew selection pin 0 : auto-deskew 1 : not auto-deskew	1'b0
[25:24]	R	RESERVED	Reserved	2'b0
[23]	R/W	DE_POL	Specifies the porality of the Data Enable (DE) (applicable to VESA and JEIDA data packing format only) 0 : High Active 1 : Low Active	1'b0
[22]	R/W	HSYNC_POL	Specifies the porality of the Horizontal Sync (HSync) (applicable to VESA and JEIDA data packing format only) – High Active means HSync is HIGH when Horizontal Sync Period 0 : High Active 1 : Low Active	1'b0
[21]	R/W	VSYNC_POL	Specifies the porality of the Verical Sync (VSync) (applicable to VESA and JEIDA data packing format only) – High Active means VSync is HIGH when Vertical Sync Period 0 : High Active 1 : Low Active	1'b0
[20:19]	R/W	LCD_FORMAT	Specifies the LVDS data packing format 0 : VESA data packing format 1 : JEIDA data packing format 2 : User Programmable data packing format 3 : Reserved (Undefined data packing format)	2'b0
[18:13]	R/W	I_LOCK_PPM_SET	PPM setting for PLL lock	6'b011011
[12:1]	R/W	I_DESKEW_CNT_SET	Adjust the period of de-skew region.	12b0110_011_1000
[0]	R/W	I_AUTO_SEL	Auto de-skew selection pin 0 : auto-deskew 1 : not auto-deskew	1'b0
LVDS Control register 1 (LVDSCTRL1)				
<i>Address : C010A004h</i>				
[31:29]	R/W	TX4010X_DUMMY	Dummy Pin (for LVDS PHY self test)	3'b0
[28]	R/W	I_ATE_MODE	Function or ATE 0 : function 1 : ATE	1'b0
[27]	R/W	I_TEST_CON_MODE	DA or I2C 0 : DA1 : I2C	1'b0
[26:24]	R/W	I_TX4010X_DUMMY	Dummy Pin (for LVDS PHY self test)	3'b0
[23:18]	R	RESERVED	Reserved	6'b0
[17:15]	R/W	SKCCK	TX output skew control pin at ODD clock ch.(Dft : 3b011)	3'b000

Bit	R/W	Symbol	Description	Reset Value
[14:12]	R/W	SKC4	TX output skew control pin at ODD ch4 (Dft : 3'b011)	3'b000
[11:9]	R/W	SKC3	TX output skew control pin at ODD ch3 (Dft : 3'b011)	3'b000
[8:6]	R/W	SKC2	TX output skew control pin at ODD ch2 (Dft : 3'b011)	3'b000
[5:3]	R/W	SKC1	TX output skew control pin at ODD ch1 (Dft : 3'b011)	3'b000
[2:0]	R/W	SKC0	TX output skew control pin at ODD ch0 (Dft : 3'b011)	3'b000
LVDS Control register 2 (LVDSCTRL2)				
<i>Address :C010 A008h</i>				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	CK_POL_SEL	Input clock polarity selection pin 0 : bypass 1 : inversion	1'b0
[14]	R/W	VSEL	VCO Frequency range selection pin 0: Low speed (40MHz ~ 90MHz) 1 : High speed (90MHz ~ 160MHz)	1'b0
[13:12]	R/W	S	Post-scaler control pin for PLL Recommend : Low speed : 2'b01, High speed : 2'b01	2'b00
[11:6]	R/W	M	Main divider control pin for PLL Recommend : Low speed : 6'b001010, High speed : 6'b001110	6'b001010
[5:0]	R/W	P	Pre-divider control pin for PLL Recommend : Low speed : 6'b001010, High speed : 6'b001110	6'b001010
LVDS Control register 3 (LVDSCTRL3)				
<i>Address :C010 A00Ch</i>				
[31:10]	R	RESERVED	Reserved	22'b0
[9:6]	R/W	SK_BIAS	Bias current control pin for Skew	4'b0011
[5]	R/W	SKEWINI	Skew selection pin 0: bypass 1: Skew enable	1'b0
[4]	R/W	SKEW_EN_H	Skew block power down 0: power down 1 : operating	1'b0
[3]	R/W	CNTB_TDLY	Delay control pin for each channel 0:25p 1:50ps	1'b0
[2]	R/W	SEL_DATABF	Input clock 1/2 division control pin	1'b0
[1:0]	R/W	SKEW_REG_CUR	Regulator bias current selection pin in SKEW block	2'b11
LVDS Control register 4 (LVDSCTRL4)				
<i>Address :C010 A010h</i>				
[31:29]	R	RESERVED	Reserved	3'b0
[28]	R/W	FLT_CNT	Filter control pin for PLL	1'b0
[27]	R/W	VOD_ONLY_CNT	the pre-emphasis's pre-driver control pin (VOD Only) 0:Disable 1 : Enable	1'b0
[26]	R/W	CNNCT_MODE_SEL	Connectivity mode selection pin 0:TX operating 1 : Connectivity check.	1'b0
[25:24]	R/W	CNNCT_CNT	Connectivity control pin 0:TX operating1: Connectivity check2,3: X (Don't care)	2'b00
[23]	R/W	VOD_HIGH_S	VOD control pin	1'b0

Bit	R/W	Symbol	Description	Reset Value
			0: normal w/ pre-emphasis 1 : Vod only	
[22]	R/W	SRC_TRH	Source termination resistor selection pin 0: off 1 : termination	1'b0
[21:14]	R/W	CNT_VOD_H	TX driver output differential voltage level control pin	8'b11111111 1
[13:6]	R/W	CNT_PEN_H	TX driver pre-emphasis level control (Dft : 8'b0000_0001)	8'b1111000 0
[5:3]	R/W	FC_CODE	Vos control pin	3'b100
[2]	R/W	OUTCON	TX Driver state selection pin 0: Hi-z 1 : Low	1'b0
[1]	R/W	LOCK_CNT	Lock signal selection pin 0 : Lock enable 1 : Lock disable	1'b0
[0]	R/W	AUTO_DSK_SEL	Auto deskew selection pin for analog 0 : Normal 1 : Auto-dekew	1'b0
LVDS LOCATION register 0 (LVDSLOC0)				
<i>Address :C010A020h</i>				
[31:4]	R	RESERVED	Reserved	28'b0
[29:24]	R/W	LOC_A4	Select a location to go in LVDS Channel A bit 4. (applicable to User Programmable data packing format only)	6'b0
[23:18]	R/W	LOC_A3	Select a location to go in LVDS Channel A bit 3. (applicable to User Programmable data packing format only)	6'b0
[17:12]	R/W	LOC_A2	Select a location to go in LVDS Channel A bit 2. (applicable to User Programmable data packing format only)	6'b0
[11:6]	R/W	LOC_A1	Select a location to go in LVDS Channel A bit 1. (applicable to User Programmable data packing format only)	6'b0
[5:0]	R/W	LOC_A0	Select a location to go in LVDS Channel A bit 0. (applicable to User Programmable data packing format only)	6'b0
LVDS LOCATION register 1 (LVDSLOC1)				
<i>Address :C010A024h</i>				
[31:4]	R	RESERVED	Reserved	28'b0
[29:24]	R/W	LOC_B2	Select a location to go in LVDS Channel B bit 2. (applicable to User Programmable data packing format only)	6'b0
[23:18]	R/W	LOC_B1	Select a location to go in LVDS Channel B bit 1. (applicable to User Programmable data packing format only)	6'b0
[17:12]	R/W	LOC_B0	Select a location to go in LVDS Channel B bit 0. (applicable to User Programmable data packing format only)	6'b0
[11:6]	R/W	LOC_A6	Select a location to go in LVDS Channel A bit 6. (applicable to User Programmable data packing format only)	6'b0
[5:0]	R/W	LOC_A5	Select a location to go in LVDS Channel A bit 5. (applicable to User Programmable data packing format only)	6'b0
LVDS LOCATION register 2 (LVDSLOC2)				
<i>Address :C010A028h</i>				
[31:4]	R	RESERVED	Reserved	28'b0
[29:24]	R/W	LOC_C0	Select a location to go in LVDS Channel C bit 0. (applicable to User Programmable data packing format only)	6'b0

Bit	R/W	Symbol	Description	Reset Value
[23:18]	R/W	LOC_B6	Select a location to go in LVDS Channel B bit 6. (applicable to User Programmable data packing format only)	6'b0
[17:12]	R/W	LOC_B5	Select a location to go in LVDS Channel B bit 5. (applicable to User Programmable data packing format only)	6'b0
[11:6]	R/W	LOC_B4	Select a location to go in LVDS Channel B bit 4. (applicable to User Programmable data packing format only)	6'b0
[5:0]	R/W	LOC_B3	Select a location to go in LVDS Channel B bit 3. (applicable to User Programmable data packing format only)	6'b0
LVDS LOCATION register 3 (LVDSLOC3)				
<i>Address :C010A02Ch</i>				
[31:4]	R	RESERVED	Reserved	28'b0
[29:24]	R/W	LOC_C5	Select a location to go in LVDS Channel C bit 5. (applicable to User Programmable data packing format only)	6'b0
[23:18]	R/W	LOC_C4	Select a location to go in LVDS Channel C bit 4. (applicable to User Programmable data packing format only)	6'b0
[17:12]	R/W	LOC_C3	Select a location to go in LVDS Channel C bit 3. (applicable to User Programmable data packing format only)	6'b0
[11:6]	R/W	LOC_C2	Select a location to go in LVDS Channel C bit 2. (applicable to User Programmable data packing format only)	6'b0
[5:0]	R/W	LOC_C1	Select a location to go in LVDS Channel C bit 1. (applicable to User Programmable data packing format only)	6'b0
LVDS LOCATION register 4 (LVDSLOC4)				
<i>Address :C010A030h</i>				
[31:4]	R	RESERVED	Reserved	28'b0
[29:24]	R/W	LOC_D3	Select a location to go in LVDS Channel D bit 3. (applicable to User Programmable data packing format only)	6'b0
[23:18]	R/W	LOC_D2	Select a location to go in LVDS Channel D bit 2. (applicable to User Programmable data packing format only)	6'b0
[17:12]	R/W	LOC_D1	Select a location to go in LVDS Channel D bit 1. (applicable to User Programmable data packing format only)	6'b0
[11:6]	R/W	LOC_D0	Select a location to go in LVDS Channel D bit 0. (applicable to User Programmable data packing format only)	6'b0
[5:0]	R/W	LOC_C6	Select a location to go in LVDS Channel C bit 6. (applicable to User Programmable data packing format only)	6'b0
LVDS LOCATION register 5 (LVDSLOC5)				
<i>Address :C010A034h</i>				
[31:4]	R	RESERVED	Reserved	28'b0
[29:24]	R/W	LOC_E1	Select a location to go in LVDS Channel E bit 1. (applicable to User Programmable data packing format only)	6'b0
[23:18]	R/W	LOC_E0	Select a location to go in LVDS Channel E bit 0. (applicable to User Programmable data packing format only)	6'b0
[17:12]	R/W	LOC_D6	Select a location to go in LVDS Channel D bit 6. (applicable to User Programmable data packing format only)	6'b0
[11:6]	R/W	LOC_D5	Select a location to go in LVDS Channel D bit 5. (applicable to User Programmable data packing format only)	6'b0
[5:0]	R/W	LOC_D4	Select a location to go in LVDS Channel D bit 4. (applicable to User Programmable data packing format only)	6'b0

Bit	R/W	Symbol	Description	Reset Value
[13]	R/W	I_BIST_FORCE_ERROR	Inserted one error into BIST transmitted pattern	1'b0
[12:7]	R/W	I_BIST_SKW_CTRL	Used the manual skew setting during data comparing in BIST 5'th bit: 0: auto-skew, 1 : manual skew control	6'b0
[6:5]	R/W	I_BIST_CLK_INV	Inverted clock during BIST operation. Bit0 is for digital and bit1 is for Analog, respectively.	2'b0
[4:3]	R/W	I_BIST_DATA_INV	Inverted the data order during BIST operation. Bit0 is for RX and bit1 is for TX, respectively.	2'b0
[2:0]	R/W	I_BIST_CH_SEL	Select the channel for BIST operation	3'b0

LVDS TEST MODE register 1 (LVDSTMODE1)

Address :C010 A054h

[31:16]	R	RESERVED	Reserved	16'b0
[15:8]	R	O_BIST_ERR_COUNT	ODD BIST error count value	
[7:3]	R	RESERVED	Reserved	5'b0
[2]	R	MON_FOR_CNNCT	Monitor pin for connectivity check	
[1]	R	O_BIST_SYNC	ODD BIST found the expected pattern and started data comparing	
[0]	R	O_BIST_STATUS	ODD Indicated whether BIST error occurs	

36.4 DisplayTop Register Summary

User uses this register to select the RGB Video data between the Primary DPC and the Secondary DPC.

Bit	R/W	Symbol	Description	Reset Value
DISPLAYTOP LVDS MUX Control register (LVDS_MUXCTRL)				
<i>Address :C010 100Ch</i>				
[31]	R/W	LVDS_MUXENB	MUX Enable 0: MUX Disable 1 : MUX Enable	1'b0
[30:2]	R/W	RESERVED	Reserved	29'b0
[1:0]	R/W	LVDS_MUXSEL	MUX Select 0: Primary DPC 1 : Secondary DPC 2-3 : Reserved (Never use this value)	2'b0

Section 37. HDMI

37.1 Overview

The HDMI (High Definition Multimedia Interface) is compatible with HDMI v1.4 spec and supports up to 1080p video resolution. It accepts RGB Video data from DPC and transmits them into HDMI cable with a serialized form.

The HDMI consists of a HDMI Link and HDMI PHY. The HDMI Link receives RGB Video data from DPC and translates them into a sequence of 10-bit signals compliant to HDMI v1.4 specification. The HDMI PHY receives a sequence of 10-bit signals from the HDMI Link and transmits them into HDMI cable with serialized form. And the HDMI PHY can generate both pixel clock and TMDS clock from the reference 24MHz clock. So users can use the generated pixel clock from the HDMI PHY, instead of external divided PLL clock from CLKGEN. DPC (Display Controller) also can use the generated pixel clock from the HDMI PHY. All the pixel clock frequency specified in HDMI v1.4 spec can be generated by the HDMI PHY.

37.1.1 Features

- HDMI 1.4a, HDCP 1.4 Complaint
- Supports Video format
 - 480p @59.94Hz/60Hz, 576p@50Hz
 - 720p @50Hz/59.94Hz/60Hz
 - 1080p @50Hz/59.94Hz/60Hz
- (not supports for interlace video format)
- Supports Color Format : 4:4:4 RGB
- Pixel Repetition : Up to x4
- Supports Bit Per Color : 8bit
- HDMI CEC (Consumer Electronics Control)
- Supports : SPDIF 2Ch, I2C 2Ch, (left/right)
- Integrated HDCP Encryption Engine for Video/Audio content protection (Authentication procedure are controlled
 - by S/W, not by H/W)
- Power down mode

Note : I2C for DDC channel is not include in the HDMI module, users must use one I2C module for DDC channel in NXP4330D/Q.

37.1.2 Block Diagram

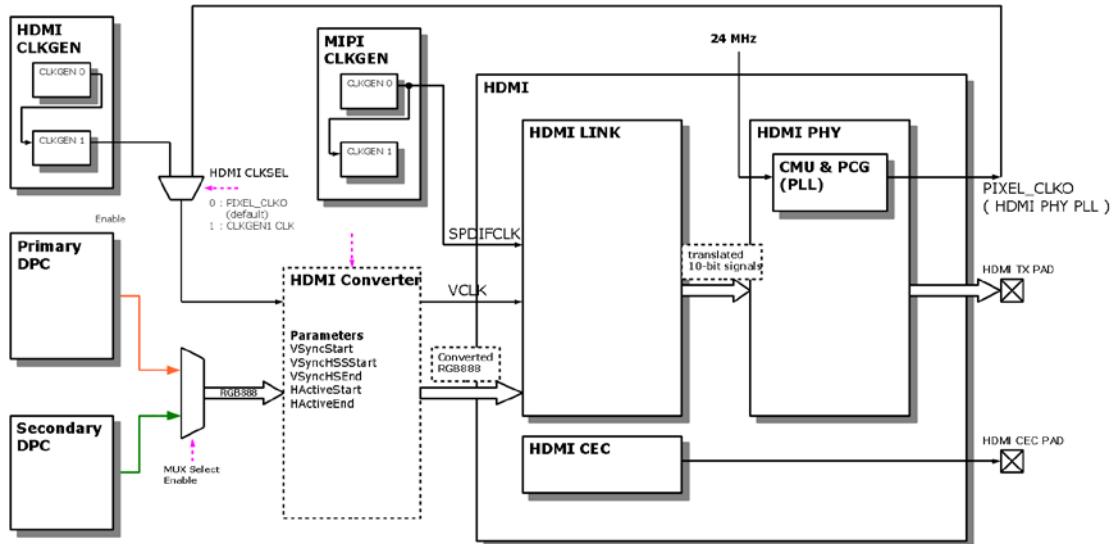


Figure 37-1. HDMI Block Diagram

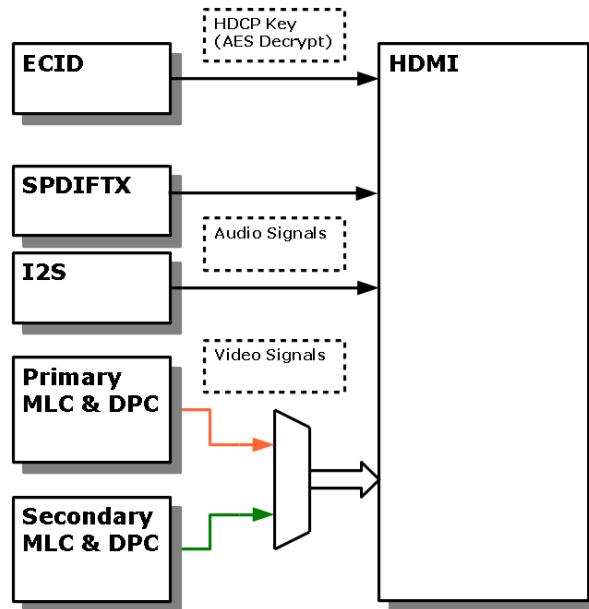


Figure 37-2. HDMI SYSTEM Block Diagram

Note : SPDIFCLK (Figure 37-1) is used to latch SDATA of I2S and SPDIFIN of SPDIFTX. SPDIFCLK frequency shall be eight times higher than that of SDATA and SPDIFIN. In SPDIF, SPDIFCLK must have higher frequency than $fs * 512$.

For example, SPDIF TX uses 48kHz frequency sampling, then SPDIFCLK must have 24.576MHz frequency. $48k * 512 = 24576k$.

37.2 Functional Description

37.2.1 Select RGB Video data for HDMI

In NXP4330D/Q, users can choose one of two RGB Video data.

- Primary DPC Video Data
- Secondary DPC Video Data

When user want to use the Primary DPC Video Data for HDMI outputs, user set the **HDMI_MUXSEL** bit of the **HDMI_MUXCTRL** register with 0 and set the **HDMI_MUXENB** bit of the **HDMI_MUXCTRL** register with 1.

37.2.2 HDMI Converter

The HDMI Converter converts the RGB Video data from DPC into a suitable format for the HDMI Link. Users must set the HDMI Converter's parameters. The HDMI Converter consists of few registers. The configuration values for the registers is in following table. The Figure 37-3 shows sync signal timing diagram.

Note : Since DPC makes sync signals, users must remember the sync information of current source video data. And users must calculate the values of V2_BLANK and V_SYNC_LINE_BEF, etc. (refered on Figure 37-3)

REGNAME	Bit	SYMBOL	Description (Refer Figure 37-3)
HDMI_SYNCCTRL0	[31]	HDMI_VCLK_SEL	This bit must be 0 (In Figure 37-1, HDMI_CLKSEL)
HDMI_SYNCCTRL0	[15:0]	HDMI_VSYNCSTART	V2_BLANK – V_SYNC_LINE_BEF_1 - 1
HDMI_SYNCCTRL1	[15:0]	HDMI_HACTIVESTART	H_BLANK - H_SYNC_START
HDMI_SYNCCTRL2	[15:0]	HDMI_HACTIVEEND	H_BLANK – H_SYNC_START + 1
HDMI_SYNCCTRL3	[31:16]	HDMI_VSYNCHSEND	H_BLANK – H_SYNC_START
HDMI_SYNCCTRL3	[15:0]	HDMI_VSYNCHSSTART	H_LINE – H_SYNC_START

Table 37-1. The configuration values for the HDMI Converter

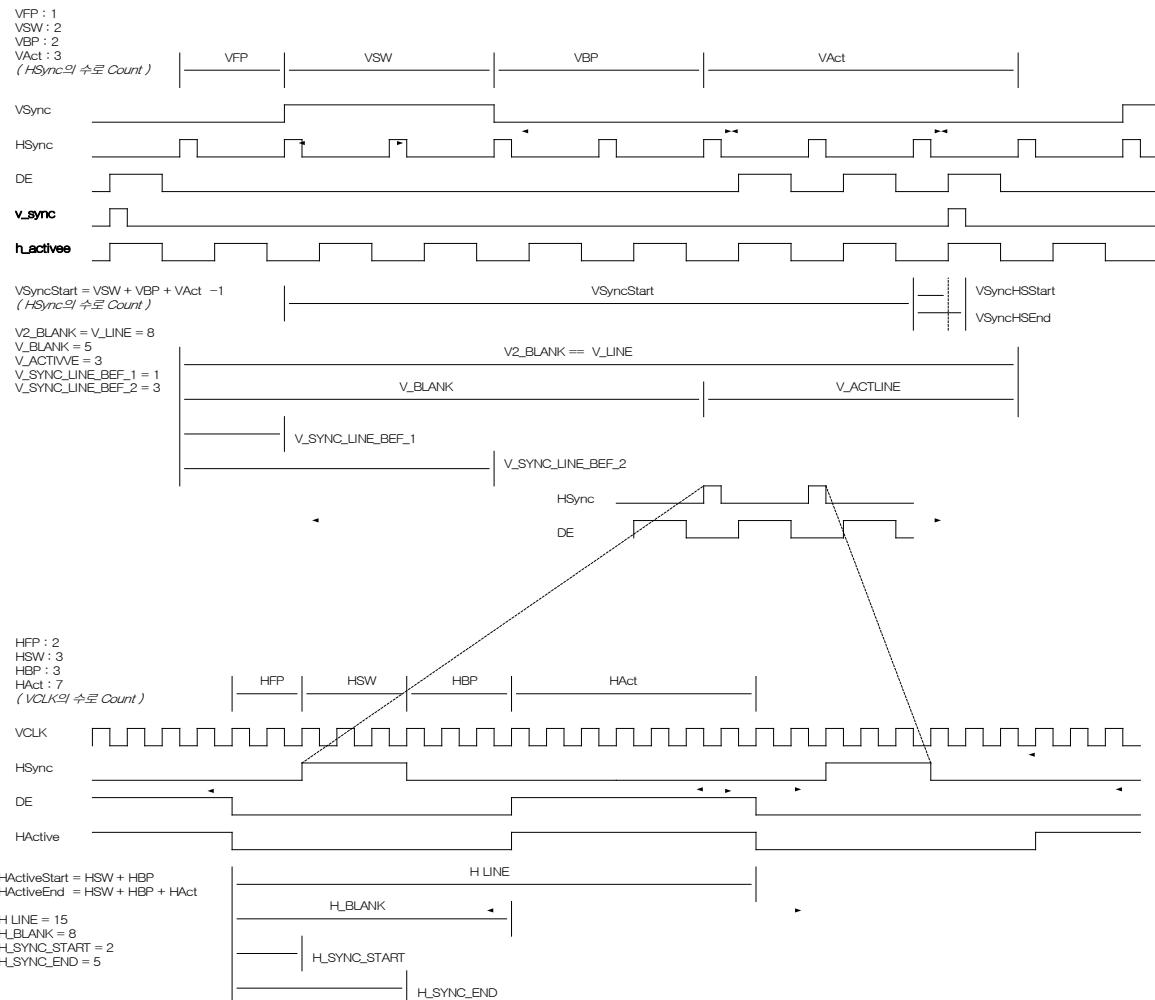


Figure 37-3. Video Sync Signal Timing Diagram

37.2.3 HDMI LINK

The HDMI Link receives a converted Video Data from the HDMI Converter. Also, the HDMI link receives a audio data from the SPDIFTX or I2S. The HDMI link translates them (video, audio) into a sequence of 10-bit signals compliant to HDMI v1.4 specification.

Users must set the configuration about video and audio.

37.2.3.1 Video Input Interface

Video input interface receives i_vsync, i_h_active, i_field and pixel data signals to identify the starting of a frame, the starting of a line and the starting of the top field in case of interlaced video formats. These signals, except for the pixel data, are different from VSYNC, HSYNC and DE signals in the CEA-861. Note that HSYNC and DE signal in CEA-861 are not used. Instead, actual vsync and hsync signals that are transferred to HDMI Rx side is generated inside hdmi_14tx_ss according to the register settings.

The i_vsync signal is activated only one cycle during active area period of the last line of the previous frame. i_h_active is always LOW a the blank period and HIGH during active period regardless of the video format. Note

that i_h_active goes HIGH even in the vertical blank period. Figure 37-4, 37-5 show the timing diagrams for the three

signals.

There are two sets of the registers related to video timing. One specifies the input timing and the other specifies the output timing. Video input/output timing with respect to the register values are also shown in Figure 37-4. Users must set the input-related registers in accordance with the input video timing. Users can use the output-related registers to control the video timing for display side, such as offset and display specific timing. Note that Figure 37-4 and 37-5 is informative and is added to illustrate the relations between register values and video timing. It is recommended to refer to register descriptions for exact timing.

Also, note that *o_vsync*, *o_hsync* and *o_r/g/b* in the figures are video timing signals described in CEA-861. These signals are generated inside *hdmi_14tx_ss* and is transferred to HDMI Rx.

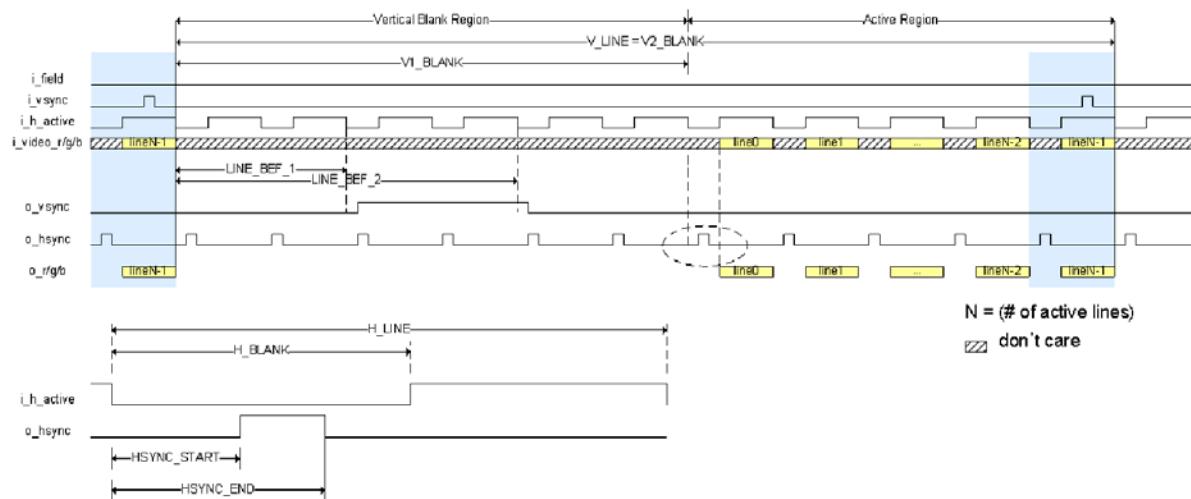


Figure 37-4. Video Timing with respect to register values in progressive mode (Informative)

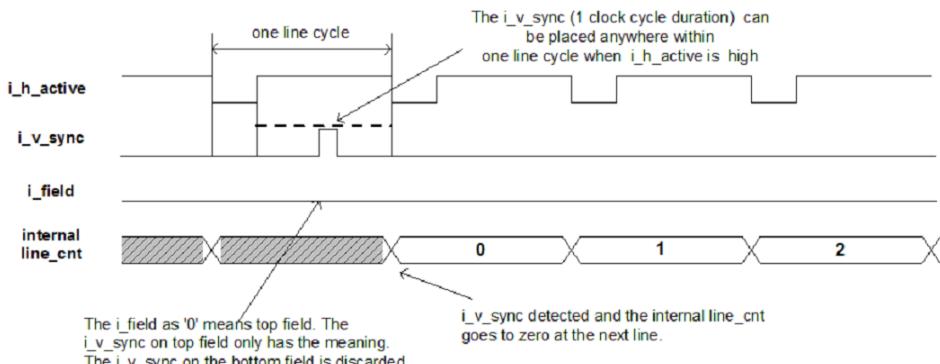


Figure 37-5. Video Input Interface timing

Below notations are used in the following explanations.

o_h_sync is equal to *o_hsync*.

o_v_sync is equal to *o_vsync*.

v_line is equal to *V_LINE*.

h_line is equal to H_LINE.
h_sync_start is equal to HSYNC_START.
h_sync_end is equal to HSYNC_END.
h_blank is equal to H_BLANK.
v1_blank is equal to V1_BLANK.
v2_blank is equal to V2_BLANK.
v_blank_f0 is equal to V_BLANK_F0.
v_blank_f1 is equal to V_BLANK_F1.
v_blank_f2 is equal to V_BLANK_F2.
v_blank_f3 is equal to V_BLANK_F3.
v_blank_f4 is equal to V_BLANK_F4.
v_blank_f5 is equal to V_BLANK_F5.
line_bef_1 is equal to LINE_BEF_1.
line_bef_2 is equal to LINE_BEF_2.
line_aft_1 is equal to LINE_AFT_1.
line_aft_2 is equal to LINE_AFT_2.
line_aft_3 is equal to LINE_AFT_3.
line_aft_4 is equal to LINE_AFT_4.
line_aft_pxl_1 is equal to LINE_AFT_PXL_1.
line_aft_pxl_2 is equal to LINE_AFT_PXL_2.
line_aft_pxl_3 is equal to LINE_AFT_PXL_3.
line_aft_pxl_4 is equal to LINE_AFT_PXL_4.
vact_space1 is equal to VACT_SPACE1.
vact_space2 is equal to VACT_SPACE2.
vact_space3 is equal to VACT_SPACE3.
vact_space4 is equal to VACT_SPACE4.
vact_space5 is equal to VACT_SPACE5.
vact_space6 is equal to VACT_SPACE6.

Below figures indicate video timing per each modes.

In 2D Progressive mode, i_field and i_vsync input video signals indicate starting of image frame. When i_vsync input video signal is active, i_field should be low.

o_v_sync is generated by line_bef_1 and line_bef_2. o_v_sync goes high at h_sync_start pixel in line_bef_1 line and goes low at h_sync_start pixel in line_bef_2 line.

v2_blank should be equal to v_line.

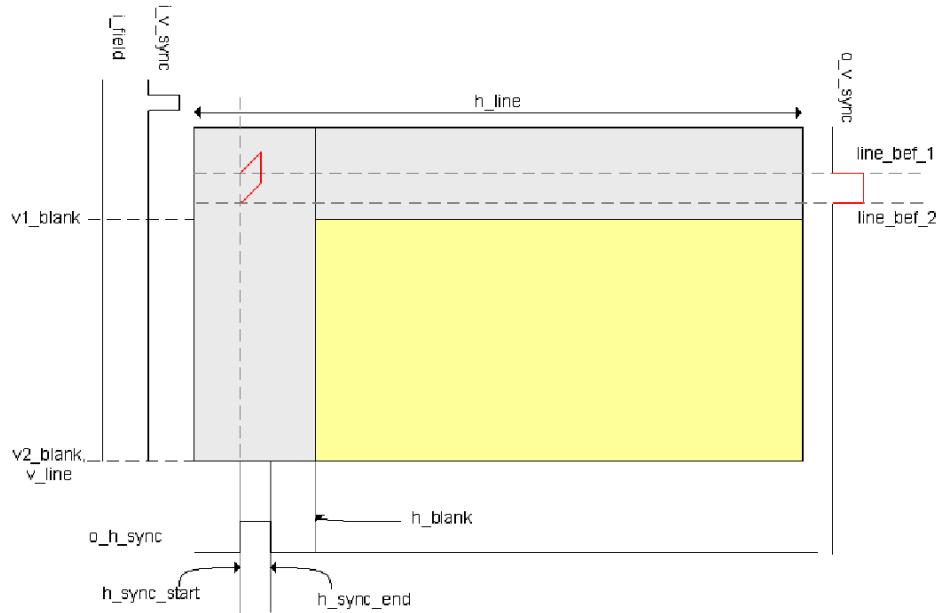


Figure 37-6. Progressive Video timing

In 3D Frame Packing Progressive mode, *i_field* and *i_vsync* input video signals indicate starting of image frame.

When *i_vsync* input video signal is active, *i_field* should be low.

o_v_sync is generated by *line_bef_1* and *line_bef_2*. *o_v_sync* is generated at *h_sync_start* in the lines.

Active space is indicated by *vact_space1* line and *vact_space2* line.

v2_blank should be equal to *v_line*.

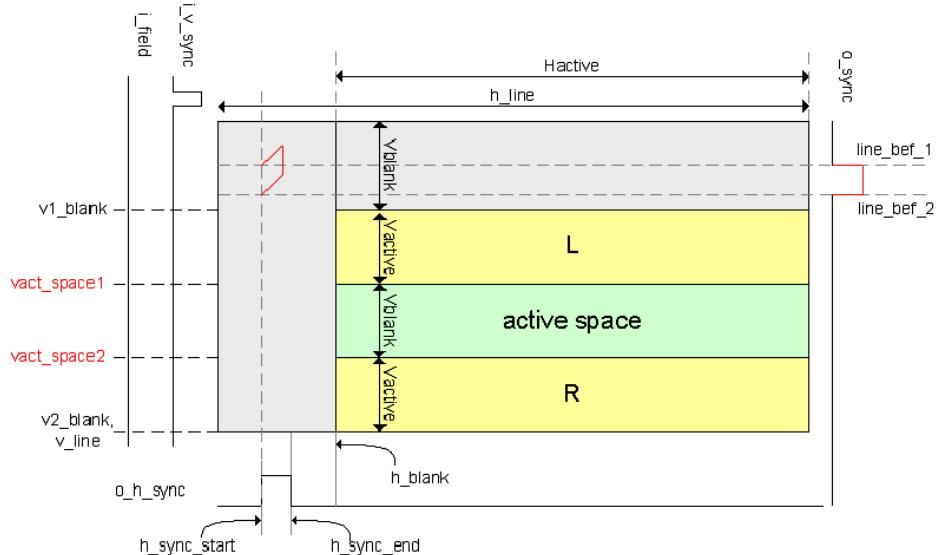


Figure 37-7. 3D Frame Packing Progressive Video timing

In 3D Side by side (half) Progressive mode, *i_field* and *i_vsync* input video signals indicate starting of image frame.

When *i_vsync* input video signal is active, *i_field* should be low.

o_v_sync is generated by *line_bef_1* and *line_bef_2*. *o_v_sync* is generated at *h_sync_start* in the lines.

v2_blank should be equal to *v_line*.

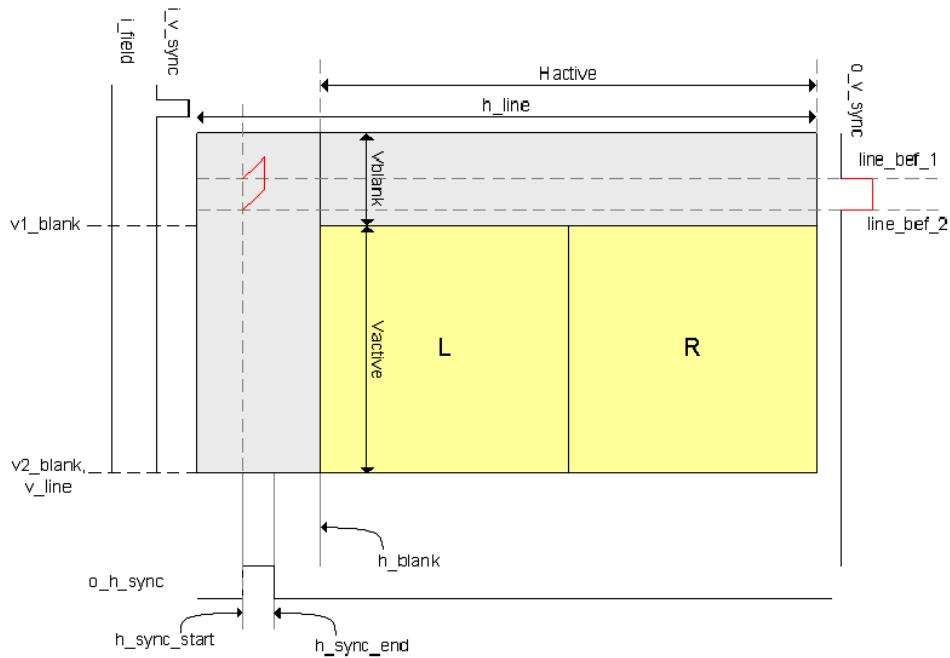


Figure 37-8. 3D Side by side (half) Progressive Video timing

In 3D Top and bottom Progressive mode, i_field and i_vsync input video signals indicate starting of image frame.

When i_vsync input video signal is active, i_field should be low.

o_v_{sync} is generated by line_bef_1 and line_bef_2. o_v_{sync} is generated at h_{sync_start} in the lines.

$v2_{blank}$ should be equal to v_{line} .

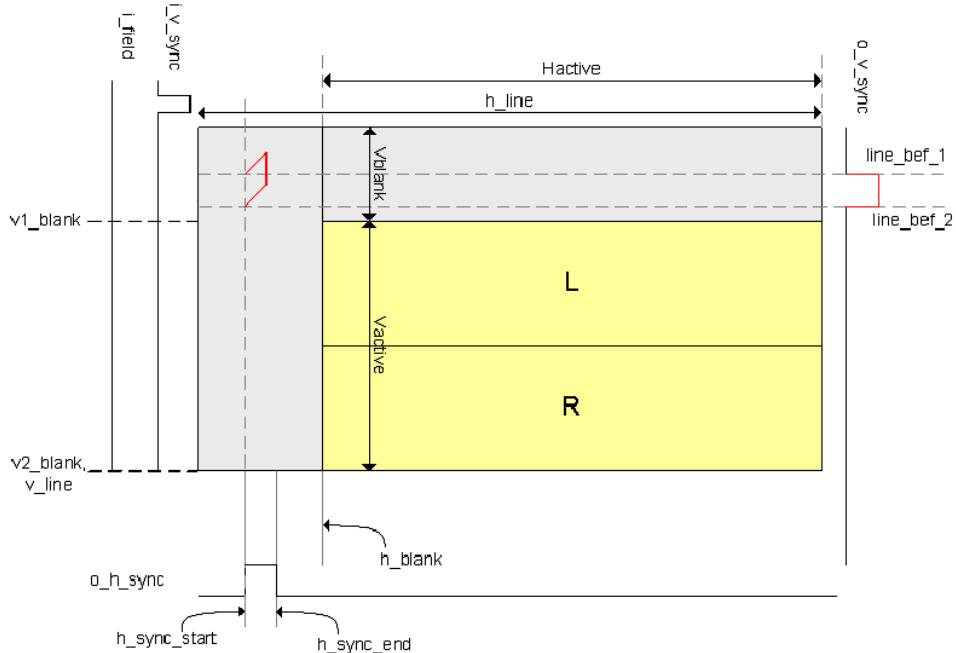


Figure 37-9. 3D Top and bottom Progressive Video timing

In 3D Line Alternative mode, i_field and i_vsync input video signals indicate starting of image frame. When i_vsync input video signal is active, i_field should be low.

o_v_{sync} is generated by $line_bef_1$ and $line_bef_2$. o_v_{sync} is generated at h_{sync_start} in the lines.

$v2_blank$ should be equal to v_line .

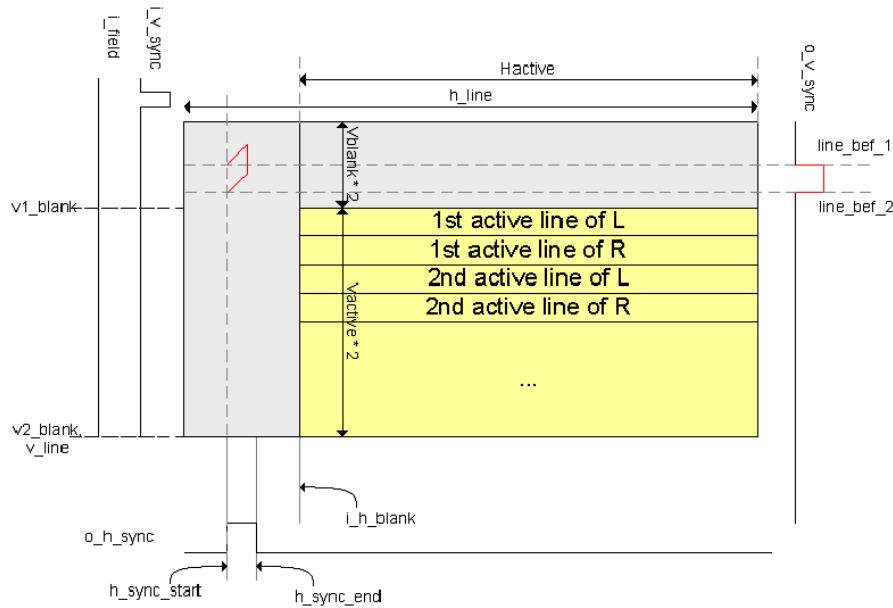


Figure 37-10. 3D Line Alternative Video timing

In 3D Side by side (full) Progressive mode, i_{field} and i_{vsync} input video signals indicate starting of image frame.

When i_{vsync} input video signal is active, i_{field} should be low.

o_v_{sync} is generated by $line_bef_1$ and $line_bef_2$. o_v_{sync} is generated at h_{sync_start} in the lines.

$v2_blank$ should be equal to v_line .

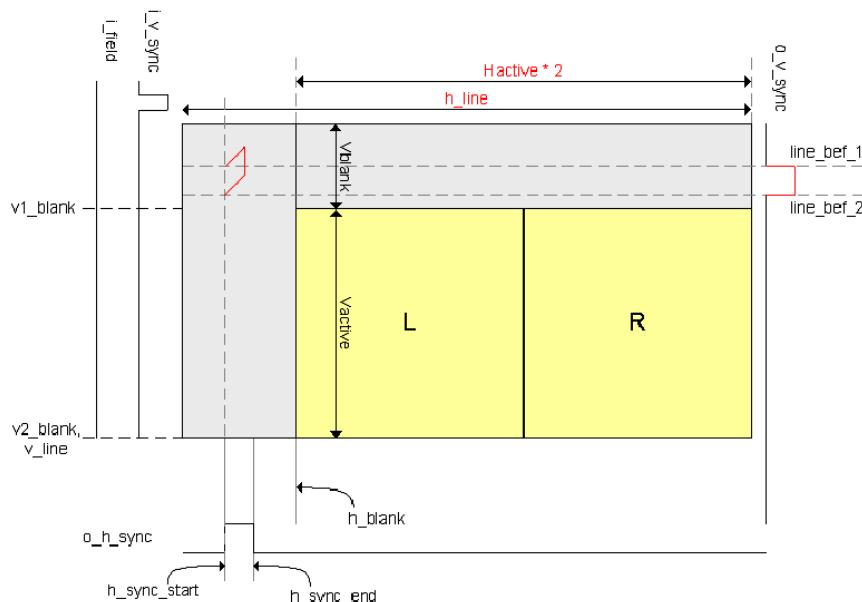


Figure 37-11. 3D Side by side (full) Progressive Video timing

In 3D L+Depth mode, i_{field} and i_{vsync} input video signals indicate starting of image frame. When i_{vsync} input

video signal is active, *i_field* should be low.

o_v_sync is generated by *line_bef_1* and *line_bef_2*. *o_v_sync* is generated at *h_sync_start* in the lines.

Active space is indicated by *vact_space1* line and *vact_space2* line.

v2_blank should be equal to *v_line*.

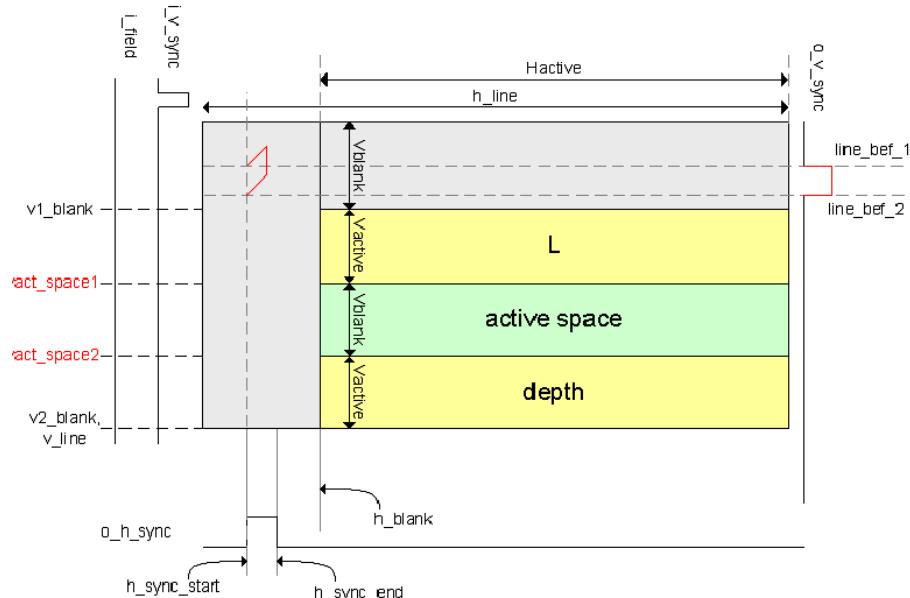


Figure 37-12. 3D L+Depth Video

37.2.3.2 Audio Input Interface

The HDMI can decrypt AES-encrypted HDCP key. For HDMI with HDCP function, each device should have a unique HDCP key set.

HDMI 1.4 Tx Subsystem supports two audio input interfaces: SPDIF Rx, and I2S Rx. Users can use SPDIF interface for two-channel Linear or Non-linear PCM Audio transmission and I2S interface for up-to eight channel Linear PCM.

For I2S interface, users can specify the channel status block and the user bit information in registers, which does not inherently exist in I2S audio format.

SPDIF Interface

HDMI 1.4 Tx Subsystem supports SPDIF Interface format that follows the IEC-60958 and IEC-61937 format. SPDIF Interface

I2S Interface

I2S interface supports linear PCM, non-linear PCM

Note : Not supported HBR audio format.

37.2.3.3 HPD

HPD signal has two transactions: rising (plugged) and falling (unplugged) transition. Users can specify the stage

number of the noise filter to reduce the possible glitches during transition.



Figure 37-13. Timing diagram for HPD unplug interrupt

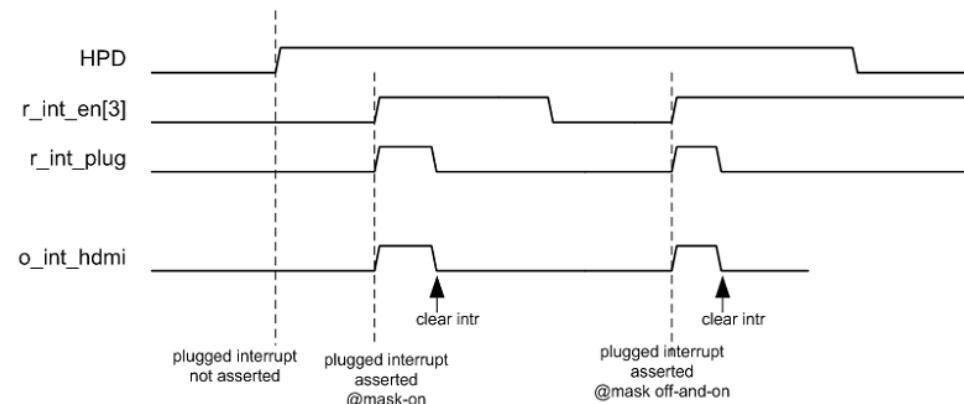


Figure 37-14. Timing diagram for HPD plug interrupt

37.2.3.4 CEC

CEC is abbreviation of Consumer Electronics Control and is used for controlling devices connected to a one-wired "CEC bus". The CEC output port is a bidirectional port, which should be pulled-up to 3.3V using external resistor and voltage supply. (i.e. an open-drain connection)

CEC devices send messages serially (MSB first) via CEC bus and the receiving device sends acknowledge bit by pulling the bus to low at ACK bit timing. For timing of each bit and structure of a message, refer to HDMI specification 1.4a, Supplement 1.(CEC specification)

37.2.3.5 Interrupt Timing

HDMI 1.4 Tx Subsystem generates level-triggered interrupts. Interrupts are masked in two levels: each submodule and controller. On the other hand, interrupt clear happens only in sub module for all interrupts except HPD. For HPD, every plug and unplug will generate, HPD plug interrupt and HPD unplug interrupt, respectively.

37.2.3.6 HDCP KEY Management

The HDMI can decrypt AES-encrypted HDCP key. For HDMI with HDCP function, each device should have a unique HDCP key set. This key set is released by Digital Contents Protection. LLC (www.digital-cp.com).

HDCP key is stored outside of the HDMI Link at a non-volatile memory. It is important that the non-volatile memory does not have original raw-HDCP. The key should be stored safely to prevent the key from eavesdroppers.

The HDMI Link uses the AES encryption algorithm to protect plain HDCP key. Before starting the HDCP authentication protocol, The HDMI Link reads encryptes HDCP keys from the memory(MEM_encr), decrypts the keys usingkeys and stores the decrypted keys to decryption memory(MEM_decr). The decryption memory can only be accessed by the HDCP engin inside the HDMI Link. The decryption memory cannot be accessed by APB3 interface to protect the

decrypted keys. The AES key is supplied via ports(i_aeskey_data, i_aeskey_hw). For the AES key, various chip id and fixed hardware value can be used. Note that only 128-bit key is supported.

The AES decryption starts by setting **AES_Start** bit of **AES_START** register. When the decryption is finished and the decrypted keys are stored in the decryption memory, the **AES_Start** bit goes to 0, notifying the decryption is done. The size of data to decrypt can be set by **AES_DATA_SIZE_H/L** register. After checking that **AES_Start** bit is 0, decrypted data can be used for HDCP key. HDCP cannot operate until the **AES_Start** bit goes to 0.

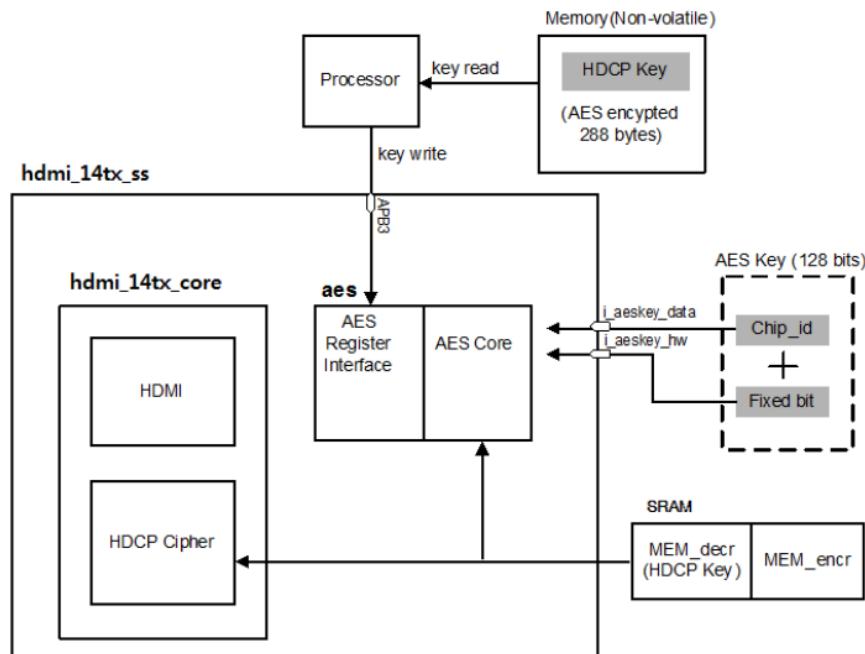


Figure 37-15. Block Diagram of HDCP Key Management

37.3 Register Summary

37.3.1 Control Registers

Register Map Summary (Base Address : 0xC020_0000)

Register	Offset	Description	Reset Value
INTC_CON_0	0x00000000	Interrupt Control Register 0	0x00
INTC_FLAG_0	0x00000004	Interrupt Flag Register 0	0x00
AESKEY_VALID	0x00000008	i_aeskey_valid value	0x0X
HPD	0x0000000C	HPD signal	0x0X
INTC_CON_1	0x00000010	Interrupt Control Register 1	0x00
INTC_FLAG_1	0x00000014	Interrupt Flag Register 1	0x0X
PHY_STATUS_0	0x00000020	PHY status Register 0	0x0X
PHY_STATUS_CMU	0x00000024	PHY CMU status Register	0xXX
PHY_STATUS_PLL	0x00000028	PHY PLL status Register	0xXX
PHY_CON_0	0x00000030	PHY Control Register	0xXX
HPD_CTRL	0x00000040	HPD Signal Control Register	0xXX
HPD_STATUS	0x00000044	HPD Status Register	0xXX
HPD_TH_X	0x00000050	HPD Status Register (HPD_TH_0~3)	0xXX

INTC_CON_0

Address = Base Address + 0x00000000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
INTRPOL	[7]	RW	Interrupt Polarity 0 : Active high 1 : Active low	
INTRENGLOBAL	[6]	RW	0 : All interrupts are disabled 1 : Interrupts are enabled or disabled by INTC_CON5:0	
RSVD	[5]	RW	Reserved	
INTRENCEC	[4]	RW	CEC interrupt enable 0 : Disabled 1 : Enabled	
INTRENHPDPLUG	[3]	RW	HPD plugged interrupt enable 0 : Disabled 1 : Enabled	
INTRENHPDUNPLUG	[2]	RW	HPD unplugged interrupt enable 0 : Disabled 1 : Enabled	

INTRENSPDIF	[1]	RW	SPDIF interrupt enable 0 : Disabled 1 : Enabled	
INTRENHDCP	[0]	RW	HDCP interrupt enable 0 : Disabled 1 : Enabled	

INTC_FLAG_0

Address = Base Address + 0x00000004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	
RSVD	[5]	RW	Reserved	
INTRCEC	[4]	RW	CEC interrupt flag. (read only) 0 : not occurred 1 : interrupt occurred	
INTRHPDPLUG	[3]	RW	HPD plugged interrupt flag. If it is written by 1, it is cleared. 0 : not occurred 1 : HPD plugged	
INTRHPDUNPLUG	[2]	RW	HPD unplugged interrupt flag. If it is written by 1, it is cleared. 0 : not occurred 1 : HPD unplugged	
INTRSPDIF	[1]	RW	SPDIF interrupt flag. (read only) 0 : not occurred 1 : interrupt occurred	
INTRHDCP	[0]	RW	HDCP interrupt flag. (read only) 0 : not occurred 1 : interrupt occurred	

AESKEY_VALID

Address = Base Address + 0x00000008, Reset Value = 0x0X

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	R	Reserved	
AESKEY_VALID	[0]	R	Reflects i_aeskey_valid signal value.*	

HPD

Address = Base Address + 0x0000000C, Reset Value = 0x0X

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	R	Reserved	

HPD_VALUE	[0]	R	Value of HPD signal 0: Unplugged 1: Plugged	
------------------	-----	---	---	--

INTC_CON_1

Address = Base Address + 0x00000010, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
INTENSINKDETECT	[1]	RW	SINK_DET interrupt enable. Triggered when SINK_DET signal from PHY goes high. INTC_CON_1[6] should also be enabled. 0 : Disabled 1 : Enabled	
INTENSINKNOTDETECT	[0]	RW	SINK_NOT_DET interrupt enable. Triggered when SINK_DET signal from PHY goes low. INTC_CON_1[6] should also be enabled. 0 : Disabled 1 : Enabled	

INTC_FLAG_1

Address = Base Address + 0x00000014, Reset Value = 0x0X

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
INTSINKDETECT	[1]	RW	SINK_DET interrupt. Triggered when SINK_DET signal from PHY goes high. INTC_CON_1[6] should also be enabled. If it is written by 1, it is cleared. 0 : not occurred 1 : SINK_DET positive edge occurred.	
INTSINKNOTDETECT	[0]	RW	SINK_NOT_DET interrupt. Triggered when SINK_DET signal from PHY goes low. INTC_CON_1[6] should also be enabled. If it is written by 1, it is cleared. 0 : not occurred 1 : SINK_DET negative edge occurred.	

PHY_STATUS_0

Address = Base Address + 0x00000020, Reset Value = 0x0X

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	R	Reserved	
SINK_DETECT	[1]	R	SINK_DET signal from PHY. 0 : Sink not detected 1 : Sink detected	
PHY_READY	[0]	R	PHY_READY signal from PHY. 0 : PHY not ready 1 : PHY ready	

PHY_STATUS_CMU

Address = Base Address + 0x00000024, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
CMU_CODE	[7:4]	R	CMU_CODE signal from PHY. CMU_CODE is the AFC(automatic frequency calibration) code that is used by the CMU to converge to the target frequency. To lock the CMU, PLL that generated TMDS clock and the pixel clock generator should be locked.	
RSVD	[3:1]	R	Reserved	
CMU_LOCK	[0]	R	CMU_LOCK signal from PHY. 0 : CMU not locked 1 : CMU locked	

PHY_STATUS_PLL

Address = Base Address + 0x00000028, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
PLL_CODE	[7:4]	R	VPLL_CODE signal from PHY. VPLL_CODE is the AFC(automatic frequency calibration) code that is used by the VPLL to converge to the target TMDS frequency.	
RSVD	[3:1]	R	Reserved	
PLL_LOCK	[0]	R	VPLL_LOCK signal from PHY. 0 : VPLL not locked 1 : VPLL locked	

PHY_CON_0

Address = Base Address + 0x00000030, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
PHY_PWR_OFF	[0]	RW	PHY power off signal. Value of this bit is propagated to o_phy_pwroff port of hdmi_14tx_ss and when connected to PHY appropriately, can power-off the PHY. Refer to PHY datasheet for more information.	

HPD_CTRL

Address = Base Address + 0x00000040, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HPD_DEGLITCH_EN	[0]	RW	Enable deglitch logic to wait for a stable HPD signal. The duration of stable signal is determined by HPD_TH_0~3 registers.	

HPD_STATUS

Address = Base Address + 0x00000044, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HPD_DEGLITCHED	[0]	RW	Current HPD signal status after deglitch logic.	

HPD_TH_x

Address = Base Address + 0x00000050, Reset Value = 0xXX

Name	Bit	Type	Description	Reset Value
HPD_TH_X	[7:0]	RW	A 32-bit HPD filter threshold value. When filter is enabled, it filters out signals stable for less than HPD_Th cycles. (based on APB cycle) Least significant byte first. For example, <ul style="list-style-type: none"> • HPD_Th[7:0] <= HPD_TH_0[7:0] • HPD_Th[31:24] <= HPD_TH_3[7:0] 	

37.3.2 Core Registers**Register Map Summary (Base Address : 0xC021_0000)**

Register	Offset	Description	Reset Value
HDMI_CON_0	0x00000000	HDMI system control register 0	0x00
HDMI_CON_1	0x00000004	HDMI system control register 1	0x00
HDMI_CON_2	0x00000008	HDMI system control register 2	0x00
STATUS	0x00000010	HDMI system status register	0x00
STATUS_EN	0x00000020	HDMI system status enable register	0x00
MODE_SEL	0x00000040	HDMI/DVI mode selection	0x00
ENC_EN	0x00000044	HDCP encryption enable register	0x00
HDMI_YMAX	0x00000060	Maximum Y (or R,G,B) pixel value	0xEB
HDMI_YMIN	0x00000064	Minimum Y (or R,G,B) pixel value	0x10
HDMI_CMAX	0x00000068	Maximum Cb/Cr pixel value	0xF0
HDMI_CMIN	0x0000006C	Minimum Cb/Cr pixel value	0x10
H_BLANK_0	0x000000A0	Horizontal blanking setting	0x00
H_BLANK_1	0x000000A4	Horizontal blanking setting	0x00
V2_BLANK_0	0x000000B0	Vertical blanking setting	0x00
V2_BLANK_1	0x000000B4	Vertical blanking setting	0x00
V1_BLANK_0	0x000000B8	Vertical blanking setting	0x00
V1_BLANK_1	0x000000BC	Vertical blanking setting	0x00
V_LINE_0	0x000000C0	vertical line setting	0x00
V_LINE_1	0x000000C4	vertical line setting	0x00
H_LINE_0	0x000000C8	Horizontal line setting	0x00
H_LINE_1	0x000000CC	Horizontal line setting	0x00
HSYNC_POL	0x000000E0	Horizontal sync polarity control register	0x00
VSYNC_POL	0x000000E4	Vertical sync polarity control register	0x00
INT_PRO_MODE	0x000000E8	Interface/Progressive control register	0x00
V_BLANK_F0_0	0x00000110	Vertical blanking setting for bottom field	0xff

V_BLANK_F0_1	0x00000114	Vertical blanking setting for bottom field	0x1f
V_BLANK_F1_0	0x00000118	Vertical blanking setting for bottom field	0xff
V_BLANK_F1_1	0x0000011C	Vertical blanking setting for bottom field	0x1f
H_SYNC_START_0	0x00000120	Horizontal sync generation setting	0x00
H_SYNC_START_1	0x00000124	Horizontal sync generation setting	0x00
H_SYNC_END_0	0x00000128	Horizontal sync generation setting	0x00
H_SYNC_END_1	0x0000012C	Horizontal sync generation setting	0x00
V_SYNC_LINE_BEF_2_0	0x00000130	Vertical sync generation for top field or frame	0xff
V_SYNC_LINE_BEF_2_1	0x00000134	Vertical sync generation for top field or frame	0x1f
V_SYNC_LINE_BEF_1_0	0x00000138	Vertical sync generation for top field or frame	0xff
V_SYNC_LINE_BEF_1_1	0x0000013C	Vertical sync generation for top field or frame	0x1f
V_SYNC_LINE_AFT_2_0	0x00000140	Vertical sync generation for bottom field - vertical position	0xff
V_SYNC_LINE_AFT_2_1	0x00000144	Vertical sync generation for bottom field - vertical position	0x1f
V_SYNC_LINE_AFT_1_0	0x00000148	Vertical sync generation for bottom field - vertical position	0xff
V_SYNC_LINE_AFT_1_1	0x0000014C	Vertical sync generation for bottom field - vertical position	0x1f
V_SYNC_LINE_AFT_PXL_2_0	0x00000150	Vertical sync generation for bottom field - horizontal position	0xff
V_SYNC_LINE_AFT_PXL_2_1	0x00000154	Vertical sync generation for bottom field - horizontal position	0x1f
V_SYNC_LINE_AFT_PXL_1_0	0x00000158	Vertical sync generation for bottom field - horizontal position	0xff
V_SYNC_LINE_AFT_PXL_1_1	0x0000015C	Vertical sync generation for bottom field - horizontal position	0x1f
V_BLANK_F2_0	0x00000160	Vertical blanking setting for third field	0xff
V_BLANK_F2_1	0x00000164	Vertical blanking setting for third field	0x1f
V_BLANK_F3_0	0x00000168	Vertical blanking setting for thrid field	0xff
V_BLANK_F3_1	0x0000016C	Vertical blanking setting for thrid field	0x1f
V_BLANK_F4_0	0x00000170	Vertical blanking setting for fourth field	0xff
V_BLANK_F4_1	0x00000174	Vertical blanking setting for fourth field	0x1f
V_BLANK_F5_0	0x00000178	Vertical blanking setting for fourth field	0xff
V_BLANK_F5_1	0x0000017C	Vertical blanking setting for fourth field	0x1f
V_SYNC_LINE_AFT_3_0	0x00000180	Vertical sync generation for third field - vertical position	0xff
V_SYNC_LINE_AFT_3_1	0x00000184	Vertical sync generation for third field - vertical position	0x1f
V_SYNC_LINE_AFT_4_0	0x00000188	Vertical sync generation for third field - vertical position	0xff
V_SYNC_LINE_AFT_4_1	0x0000018C	Vertical sync generation for third field - vertical position	0x1f
V_SYNC_LINE_AFT_5_0	0x00000190	Vertical sync generation for fourth field - vertical position	0xff
V_SYNC_LINE_AFT_5_1	0x00000194	Vertical sync generation for fourth field - vertical position	0x1f
V_SYNC_LINE_AFT_6_0	0x00000198	Vertical sync generation for fourth field - vertical position	0xff
V_SYNC_LINE_AFT_6_1	0x0000019C	Vertical sync generation for fourth field - vertical position	0x1f
V_SYNC_LINE_AFT_PXL_3_0	0x000001A0	Vertical sync generation for third field - horizontal position	0xff
V_SYNC_LINE_AFT_PXL_3_1	0x000001A4	Vertical sync generation for third field - horizontal position	0x1f

V_SYNC_LINE_AFT_PXL_4_0	0x000001A8	Vertical sync generation for third field - horizontal position	0xff
V_SYNC_LINE_AFT_PXL_4_1	0x000001AC	Vertical sync generation for third field - horizontal position	0x1f
V_SYNC_LINE_AFT_PXL_5_0	0x000001B0	Vertical sync generation for fourth field - horizontal position	0xff
V_SYNC_LINE_AFT_PXL_5_1	0x000001B4	Vertical sync generation for fourth field - horizontal position	0x1f
V_SYNC_LINE_AFT_PXL_6_0	0x000001B8	Vertical sync generation for fourth field - horizontal position	0xff
V_SYNC_LINE_AFT_PXL_6_1	0x000001BC	Vertical sync generation for fourth field - horizontal position	0x1f
VACT_SPACE1_0	0x000001C0	1st Vertical Active Space start line	0xff
VACT_SPACE1_1	0x000001C4	1st Vertical Active Space end line	0x1f
VACT_SPACE2_0	0x000001C8	1st Vertical Active Space start line	0xff
VACT_SPACE2_1	0x000001CC	1st Vertical Active Space end line	0x1f
VACT_SPACE3_0	0x000001D0	2nd Vertical Active Space start line	0xff
VACT_SPACE3_1	0x000001D4	2nd Vertical Active Space end line	0x1f
VACT_SPACE4_0	0x000001D8	2nd Vertical Active Space start line	0xff
VACT_SPACE4_1	0x000001DC	2nd Vertical Active Space end line	0x1f
VACT_SPACE5_0	0x000001E0	3rd Vertical Active Space start line	0xff
VACT_SPACE5_1	0x000001E4	3rd Vertical Active Space end line	0x1f
VACT_SPACE6_0	0x000001E8	3rd Vertical Active Space start line	0xff
VACT_SPACE6_1	0x000001EC	3rd Vertical Active Space end line	0x1f
GCP_CON	0x00000200	GCP packet control register	0x04
GCP_BYTE1	0x00000210	GCP packet body	0x00
GCP_BYTE2	0x00000214	GCP packet body	0x00
GCP_BYTE3	0x00000218	GCP packet body	0x00
ASP_CON	0x00000300	ASP packet control register	0x00
ASP_SP_FLAT	0x00000304	ASP packet sp_flat bit control	0x00
ASP_CHCFG0	0x00000310	ASP audio channel configuration	0x08
ASP_CHCFG1	0x00000314	ASP audio channel configuration	0x1A
ASP_CHCFG2	0x00000318	ASP audio channel configuration	0x2C
ASP_CHCFG3	0x0000031C	ASP audio channel configuration	0x3E
ACR_CON	0x00000400	ACR packet control register	0x00
ACR_MCTS0	0x00000410	Measured CTS value	0x01
ACR_MCTS1	0x00000414	Measured CTS value	0x00
ACR_MCTS2	0x00000418	Measured CTS value	0x00
ACR_N0	0x00000430	N value for ACR packet	0xE8
ACR_N1	0x00000434	N value for ACR packet	0x03
ACR_N2	0x00000438	N value for ACR packet	0x00
ACP_CON	0x00000500	ACP packet control register	0x00
ACP_TYPE	0x00000514	ACP packet header	0x00

ACP_DATAX	0x00000520	ACP packet body	0x00
ISRC_CON	0x00000600	ACR packet control register	0x00
ISRC1_HEADER1	0x00000614	ISCR1 packet header	0x00
ISRC1_DATAX	0x00000620	ISRC1 packet body	0x00
ISRC2_DATAX	0x000006A0	ISRC2 packet body	0x00
AVI_CON	0x00000700	AVI packet control register	0x00
AVI_HEADER0	0x00000710	AVI packet header	0x00
AVI_HEADER1	0x00000714	AVI packet header	0x00
AVI_HEADER2	0x00000718	AVI packet header	0x00
AVI_CHECK_SUM	0x0000071C	AVI packet checksum	0x00
AVI_BYTEX	0x00000720	AVI packet body	0x00
AUI_CON	0x00000800	AUI packet control register	0x00
AUI_HEADER0	0x00000810	AUI packet header	0x00
AUI_HEADER1	0x00000814	AUI packet header	0x00
AUI_HEADER2	0x00000818	AUI packet header	0x00
AUI_CHECK_SUM	0x0000081C	AUI packet checksum	0x00
AUI_BYTEX	0x00000820	AUI packet body	0x00
MPG_CON	0x00000900	ACR packet control register	0x00
MPG_CHECK_SUM	0x0000091C	MPG packet checksum	0x00
MPG_DATAX	0x00000920	MPG packet body	0x00
SPD_CON	0x00000A00	SPD packet control register	0x00
SPD_HEADER0	0x00000A10	SPD packet header	0x00
SPD_HEADER1	0x00000A14	SPD packet header	0x00
SPD_HEADER2	0x00000A18	SPD packet header	0x00
SPD_DATAX	0x00000A20	SPD packet body	0x00
GAMUT_CON	0x00000B00	GAMUT packet control register	0x00
GAMUT_HEADER0	0x00000B10	GAMUT packet header	0x00
GAMUT_HEADER1	0x00000B14	GAMUT packet header	0x00
GAMUT_HEADER2	0x00000B18	GAMUT packet header	0x00
GAMUT_METADATAX	0x00000B20	GAMUT packet body	0x00
VSI_CON	0x00000C00	VSI packet control register	0x00
VSI_HEADER0	0x00000C10	VSI packet header	0x00
VSI_HEADER1	0x00000C14	VSI packet header	0x00
VSI_HEADER2	0x00000C18	VSI packet header	0x00
VSI_DATAX	0x00000C20	VSI packet body	0x00
DC_CONTROL	0x00000D00	Deep Color Control Register	0x00
VIDEO_PATTERN_GEN	0x00000D04	Video Pattern Generation Register	0x00

AN_SEED_SEL	0x00000E48	An seed selection register.	0xFF
AN_SEED_0	0x00000E58	An seed value register	0x00
AN_SEED_1	0x00000E5C	An seed value register	0x00
AN_SEED_2	0x00000E60	An seed value register	0x00
AN_SEED_3	0x00000E64	An seed value register	0x00
HDCP_SHA1_X	0x00007000	SHA-1 value from repeater	0x00
HDCP_KSV_LIST_X	0x00007050	KSV list from repeater	0x00
HDCP_KSV_LIST_CON	0x00007064	KSV list control	0x00
HDCP_SHA_RESULT	0x00007070	SHA-1 checking result register	0x00
HDCP_CTRL1	0x00007080	HDCP control register1	0x00
HDCP_CTRL2	0x00007084	HDCP control register2	0x00
HDCP_CHECK_RESULT	0x00007090	Ri and Pj value checking result	0x00
HDCP_BKSV_X	0x000070A0	KSV of Rx	0x00
HDCP_AKSV_X	0x000070C0	KSV of Tx	0x00
HDCP_AN_X	0x000070E0	An value	0x00
HDCP_BCAPS	0x00007100	BCAPS from Rx	0x00
HDCP_BSTATUS_0	0x00007110	BSTATUS from Rx	0x00
HDCP_BSTATUS_1	0x00007114	BSTATUS from Rx	0x00
HDCP_RI_0	0x00007140	Ri value of Tx	0x00
HDCP_RI_1	0x00007144	Ri value of Tx	0x00
HDCP_I2C_INT	0x00007180	I2C interrupt flag	0x00
HDCP_AN_INT	0x00007190	An value ready interrupt flag	0x00
HDCP_WATCGDOG_INT	0x000071A0	Wachdog interrupt flag	0x00
HDCP_RI_INT	0x000071B0	Ri value update interrupt flag	0x00
HDCP_RI_COMPARE_0	0x000071D0	HDCP Ri Interrupt Frame number index register 0	0x80
HDCP_RI_COMPARE_1	0x000071D4	HDCP Ri Interrupt Frame number index register 1	0x7F
HDCP_FRAME_COUNT	0x000071E0	Current value of the frame count index in the hardware	0x00
RGB_ROUND_EN	0x0000D500	round enable for 8/10 bit R/G/B in video_receiver	0x00
VACT_SPACE_R_0	0x0000D504	vertical active space R	0x00
VACT_SPACE_R_1	0x0000D508	vertical active space R	0x00
VACT_SPACE_G_0	0x0000D50C	vertical active space G	0x00
VACT_SPACE_G_1	0x0000D510	vertical active space G	0x00
VACT_SPACE_B_0	0x0000D514	vertical active space B	0x00
VACT_SPACE_B_1	0x0000D518	vertical active space B	0x00
BLUE_SCREEN_R_0	0x0000D520	R Pixel values for blue screen [3:0]	0x00
BLUE_SCREEN_R_1	0x0000D524	R Pixel values for blue screen [11:4]	0x00
BLUE_SCREEN_G_0	0x0000D528	G Pixel values for blue screen [3:0]	0x00

BLUE_SCREEN_G_1	0x0000D52C	G Pixel values for blue screen [11:4]	0x00
BLUE_SCREEN_B_0	0x0000D530	B Pixel values for blue screen [3:0]	0x00
BLUE_SCREEN_B_1	0x0000D534	B Pixel values for blue screen [11:4]	0x00

HDMI_CON_0

Address = Base Address + 0x00000000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	
BLUE_SCR_EN	[5]	RW	Blue screen mode control. When set, the input video pixels are discarded and BLUESCREEN register values are transmitted for all video data period. 0: Disable 1: Enable	
ENCODING_OPTION	[4]	RW	10-bit TMDS encoding bit order option 0: bit order reverse among the 10-bit encoding (to be set to 1 when connecting to the TMDS PHY 1.3) 1: bit order as it is	
YCBCR422_SEL	[3]	RW	Video Input mode control. 0: 4:4:4 mode 1: 4:2:2 12 bit YCbCr When 8-bit mode, the 12-bit inputs are rounded up to generate 8-bit outputs.	
ASP_E	[2]	RW	Audio sample packet generation control. This bit is only valid when SYSTEM_EN is set. 0: discard audio sample 1: When the audio sample is received, the audio sample packet is generated.	
POWER_DOWN	[1]	RW	TMDS PHY power down mode. When it's set to 0, data could not be transferred to a receiver. 0: normal operation mode 1: power down	
SYSTEM_EN	[0]	RW	HDMI system enable. 0: No op. 1: HDMI enable	

HDMI_CON_1

Address = Base Address + 0x00000004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	
PXL_LMT_CTRL	[6:5]	RW	Pixel value limitation control 0b00 : By-pass (Do not limit the pixel value) 0b01 : RGB mode All channel's video input pixels are limited according to YMAX and YMIN register values. 0b10: YCbCr mode The value of I_VIDEO_G is limited according to YMAX and YMIN. The values of I_VIDEO_B and I_VIDEO_R are limited according to CMAX and CMIN. 0b11 : Reserved	

RSVD	[4:2]	RW	Reserved	
PXL_REP_RATIO	[1:0]	RW	Pixel repetition ratio 0b00 : no pixel repetition 0b01 : 2 times repetition 0b10 : 3 times repetition 0b11 : 4 times repetition * Use the resulting video mode setting for pixel repetition. e.g. For 720x480p (pixel repetition = 1) Use 1440x480p video mode	

HDMI_CON_2

Address = Base Address + 0x00000008, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	
VID_PERIOD_EN	[5]	RW	Video preamble control. 0 : Video Preamble is applied. (HDMI mode) 1 : Video Preamble is not applied (DVI mode)	
RSVD	[4:2]	RW	Reserved	
DVI_BAND_EN	[1]	RW	In DVI mode, the leading guard band is not used. 0 : Guard band is applied (HDMI mode) 1 : Guard band is not applied (DVI mode)	
RSVD	[0]	RW	Reserved	

STATUS

Address = Base Address + 0x00000010, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUTHEN_ACK	[7]	RW	When hdcp is authenticated, it occurs. This bit keeps the authentication signal constantly. It's not cleared at all. It's just one delayed signal of authen_ack signal from hdcp block. This bit is not an interrupt source. Read Only bit 0: not authenticated 1: authenticated	
AUD_FIFO_OVF	[6]	RW	When audio FIFO is overflowed, this bit will be set. Once it is set, it should be cleared by host. 0: not full 1: full	
RSVD	[5]	RW	Reserved	
UPDATE_RI_INT	[4]	RW	Ri Interrupt status bit If it is written by 1, it is cleared. 0: not occurred 1: interrupt occurred	
RSVD	[3]	RW	Reserved	
AN_WRITE_INT	[2]	RW	\$\$\$ 비어있음	

WATCHDOG_INT	[1]	RW	Indicates that 2nd part of HDCP authentication protocol is initiated and CPU should set a watchdog timer to check 5 sec interval. If it is written by 1, it is cleared. 0: not occurred 1: interrupt occurred	
I2C_INIT_INT	[0]	RW	Indicates that 1st part of HDCP authentication protocol can start. If it is written by 1, it is cleared. 0: not occurred 1: interrupt occurred	

STATUS_EN

Address = Base Address + 0x00000020, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	
AUD_FIDO_OVF_EE	[6]	RW	Audio buffer overflow interrupt enable When it is set to 1, interrupt assertion is written on the STATUS registers. 0 : Disable 1 : Enable	
RSVD	[5]	RW	Reserved	
UPDATE_RI_INT_EN	[4]	RW	UPDATE_RI_INT interrupt enable. 0 : Disable 1 : Enable	
RSVD	[3]	RW	Reserved	
AN_WRITE_INT_EN	[2]	RW	AN_WRITE_INT interrupt enable. 0 : Disable 1 : Enable	
WATCHDOG_INT_EN	[1]	RW	WATCHDOG_INT interrupt enable. 0 : Disable 1 : Enable	
I2C_INT_EN	[0]	RW	I2C_INT interrupt enable. 0 : Disable 1 : Enable	

MODE_SEL

Address = Base Address + 0x00000040, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
HDMI_MODE	[1]	RW	Select a mode. 0 : Disable 1 : Enable	
DVI_MODE	[0]	RW	Select a mode. 0 : Disable 1 : Enable	

ENC_EN

Address = Base Address + 0x00000044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HDCP_ENC_EN	[0]	RW	When set, HDCP encryption is applied. Before setting this bit, the HDCP authentication process has to be accomplished. 0: Encryption disable 1: Enable	

HDMI_YMAX

Address = Base Address + 0x00000060, Reset Value = 0xEB

Name	Bit	Type	Description	Reset Value
HDMI_YMAX	[7:0]	RW	Maximum value of Y(or G for RGB format) value after pixel limit control. These registers are used according to Pxl_Lmt_Ctrl bits in HDMI_CON_1 register. When Pxl_Lmt_Ctrl bits is 0x1, R and B for RGB format is also limited by the value of this register. For RGB mode if (i_video_x >= HDMI_YMAX x 16) output = HDMI_YMAX x 16 else if (i_video_x < HDMI_YMIN x 16) output = HDMI_YMIN x 16 else output = i_video_x For YCbCr mode, the Y input is dealt with the same as above. For Cb and Cr values, if (i_video_x >= HDMI_CMAX x 16) output = HDMI_CMAX x 16 else if (i_video_x < HDMI_CMIN x 16) output = HDMI_CMIN x 16 else output = i_video_x Note) Value 16 in each line compensates the difference of bit width between the input pixel and register value.	

HDMI_YMIN

Address = Base Address + 0x00000064, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value

HDMI_YMIN	[7:0]	RW	<p>Minimum value of Y(or G for RGB format) value after pixel limit control.</p> <p>These registers are used according to Pxl_Lmt_Ctrl bits in HDMI_CON_1 register.</p> <p>When Pxl_Lmt_Ctrl bits is 0x1, R and B for RGB format is also limited by the value of this register.</p> <p>For RGB mode</p> <pre>if (i_video_x >= HDMI_YMAX x 16) output = HDMI_YMAX x 16 else if (i_video_x < HDMI_YMIN x 16) output = HDMI_YMIN x 16 else output = i_video_x</pre> <p>For YCbCr mode, the Y input is dealt with the same as above. For Cb and Cr values,</p> <pre>if (i_video_x >= HDMI_CMAX x 16) output = HDMI_CMAX x 16 else if (i_video_x < HDMI_CMIN x 16) output = HDMI_CMIN x 16 else output = i_video_x</pre> <p>Note) Value 16 in each line compensates the difference of bit width between the input pixel and register value.</p>	
-----------	-------	----	---	--

HDMI_CMAX

Address = Base Address + 0x00000068, Reset Value = 0xF0

Name	Bit	Type	Description	Reset Value
HDMI_CMAX	[7:0]	RW	<p>Maximum value of Cb and Cr(or R and B for RGB format) value after pixel limit control.</p> <p>These registers are used according to Pxl_Lmt_Ctrl bits in HDMI_CON_1 register.</p> <p>When Pxl_Lmt_Ctrl bits is 0x2, R and B for RGB format is also limited by the value of this register.</p> <p>If Pxl_Lmt_Ctrl bits is 0x1, R and B for RGB format is also limited by</p> <p>HDMI_YMAX register. For RGB mode</p> <pre>if (i_video_x >= HDMI_YMAX x 16) output = HDMI_YMAX x 16 else if (i_video_x < HDMI_YMIN x 16) output = HDMI_YMIN x 16 else output = i_video_x</pre> <p>For YCbCr mode, the Y input is dealt with the same as above. For Cb and Cr values,</p> <pre>if (i_video_x >= HDMI_CMAX x 16) output = HDMI_CMAX x 16 else if (i_video_x < HDMI_CMIN x 16) output = HDMI_CMIN x 16 else output = i_video_x</pre> <p>Note) Value 16 in each line compensates the difference of bit width between the input pixel and register value.</p>	

HDMI_CMIN

Address = Base Address + 0x0000006C, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value

HDMI_CMIN	[7:0]	RW	<p>Minimum value of Cb and Cr(or R and B for RGB format) value after pixel limit control.</p> <p>These registers are used according to Pxl_Lmt_Ctrl bits in HDMI_CON_1 register.</p> <p>When Pxl_Lmt_Ctrl bits is 0x2, R and B for RGB format is also limited by the value of this register.</p> <p>If Pxl_Lmt_Ctrl bits is 0x1, R and B for RGB format is also limited by HDMI_YMIN register. For RGB mode</p> <pre> if (i_video_x >= HDMI_YMAX * 16) output = HDMI_YMAX * 16 else if (i_video_x < HDMI_YMIN * 16) output = HDMI_YMIN * 16 else output = i_video_x </pre> <p>For YCbCr mode, the Y input is dealt with the same as above. For Cb and Cr values,</p> <pre> if (i_video_x >= HDMI_CMAX * 16) output = HDMI_CMAX * 16 else if (i_video_x < HDMI_CMIN * 16) output = HDMI_CMIN * 16 else output = i_video_x </pre> <p>Note) Value 16 in each line compensates the difference of bit width between the input pixel and register value.</p>
-----------	-------	----	--

H_BLANK_0

Address = Base Address + 0x000000A0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
H_BLANK	[7:0]	RW	H_BLANK [7:0] of 13 bits. Clock Cycles of horizontal Blanking Size. Refer to the Reference CEA-861D	

H_BLANK_1

Address = Base Address + 0x000000A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
H_BLANK	[4:0]	RW	H_BLANK [12:8] of 13 bits. Clock Cycles of horizontal Blanking Size. Refer to the Reference CEA-861D	

V2_BLANK_0

Address = Base Address + 0x000000B0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
V2_BLANK	[7:0]	RW	V2_BLANK [7:0] of 13 bits. V1_BLANK+Active Lines. End Part. This value is the same as V_LINE value for progressive mode. For interlace mode, use the reference value in the table. Refer to the Reference CEA-861D	

V2_BLANK_1

Address = Base Address + 0x000000B4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	

V2_BLANK	[4:0]	RW	V2_BLANK[12:8] of 13 bits. V1_BLANK+Active Lines. End Part. This value is the same as V_LINE value for progressive mode. For interlace mode, use the reference value in the table. Refer to the Reference CEA-861D	
----------	-------	----	---	--

V1_BLANK_0

Address = Base Address + 0x000000B8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
V1_BLANK	[7:0]	RW	V1_BLANK[7:0] of 13 bits. Vertical Blanking Line Size. Front Part. Refer to the Reference CEA-861D	

V1_BLANK_1

Address = Base Address + 0x000000BC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V1_BLANK	[4:0]	RW	V1_BLANK[12:8] of 13 bits. Vertical Blanking Line Size. Front Part. Refer to the Reference CEA-861D	

V_LINE_0

Address = Base Address + 0x000000C0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
V_LINE	[7:0]	RW	V_LINE[7:0] of 13 bits. Vertical Line Length. Refer to the Reference CEA-861D	

V_LINE_1

Address = Base Address + 0x000000C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_LINE	[4:0]	RW	V_LINE[12:8] of 13 bits. Vertical Line Length. Refer to the Reference CEA-861D	

H_LINE_0

Address = Base Address + 0x000000C8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
H_LINE	[7:0]	RW	H_LINE[7:0] of 13 bits. Horizontal Line Length. Refer to the Reference CEA-861D	

Reserved

H_LINE_1

Address = Base Address + 0x000000CC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	R	Reserved	
H_LINE	[4:0]	RW	H_LINE [12:8] of 13 bits. Horizontal Line Length. Refer to the Reference CEA-861D	

HSYNC_POL

Address = Base Address + 0x000000E0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HSYNC_POL	[0]	RW	Set this bit for inverting the generated signal to meet the modes. In 720p and 1080i modes don't need to invert the signal. Others need to be inverted. Refer to the Reference CEA-861D 0 : active high 1 : active low 0 : active high 1 : active low	

VSYNC_POL

Address = Base Address + 0x000000E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
V_SYNC_POL_SEL	[0]	RW	Start point detection polarity selection bit. 720p or 1080i's sync shapes are different from 480p,480i, and 576p's. They are inverted shapes. 0 : active high 1 : active low	

INT_PRO_MODE

Address = Base Address + 0x000000E8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
INT_PRO_MODE	[0]	RW	Interlaced or Progressive Mode Selection. Refer to the Reference CEA-861D 0: progressive 1: interlaced.	

V_BLANK_F0_0

Address = Base Address + 0x00000110, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value

V_BLANK_F0	[7:0]	RW	v_blank_f0 [7:0] of 13 bits. The start position of bottom field's active region. This value is the same as V_LINE value for Interlace mode. For progressive mode, This value is not used. Refer to the Reference CEA-861D	
------------	-------	----	--	--

V_BLANK_F0_1

Address = Base Address + 0x000000114, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_BLANK_F0	[4:0]	RW	v_blank_f0 [12:8] of 13 bits. The start position of bottom field's active region. This value is the same as V_LINE value for Interlace mode. For progressive mode, This value is not used. Refer to the Reference CEA-861D	

V_BLANK_F1_0

Address = Base Address + 0x000000118, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_BLANK_F1	[7:0]	RW	v_blank_f1 [7:0] of 13 bits. In the interface mode, v_blank length of even field and odd field is different. This register specifies the end position of bottom field's active region. Refer to the Reference CEA-861D	

V_BLANK_F1_1

Address = Base Address + 0x00000011C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_BLANK_F1	[4:0]	RW	v_blank_f1 [12:8] of 13 bits. In the interface mode, v_blank length of even field and odd field is different. This register specifies the end position of bottom field's active region. Refer to the Reference CEA-861D	

H_SYNC_START_0

Address = Base Address + 0x000000120, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HSYNC_START	[7:0]	RW	Hsync_Start [7:0] of 11 bits. Set the start point of H sync. Refer to the Reference CEA-861D	

H_SYNC_START_1

Address = Base Address + 0x000000124, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
HSYNC_START	[4:0]	RW	Hsync_Start [12:8] of 11 bits. Set the start point of H sync. Refer to the Reference CEA-861D	

H_SYNC_END_0

Address = Base Address + 0x00000128, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HSYNC_EDN	[7:0]	RW	Hsync_Edn [7:0] of 11 bits. Set the end point of H sync. Refer to the Reference CEA-861D	

H_SYNC_END_1

Address = Base Address + 0x0000012C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
HSYNC_EDN	[4:0]	RW	Hsync_Edn [12:8] of 11 bits. Set the end point of H sync. Refer to the Reference CEA-861D	

V_SYNC_LINE_BEF_2_0

Address = Base Address + 0x00000130, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_BEF_2	[7:0]	RW	v_sync_line_bef_2 [7:0] of 13 bits. Top field (or frame) V sync end line number. Refer to the Reference CEA-861D	

V_SYNC_LINE_BEF_2_1

Address = Base Address + 0x00000134, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_BEF_2	[4:0]	RW	v_sync_line_bef_2 [12:8] of 13 bits. Top field (or frame) V sync end line number. Refer to the Reference CEA-861D	

V_SYNC_LINE_BEF_1_0

Address = Base Address + 0x00000138, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_BEF_1	[7:0]	RW	Top field (or frame) V sync start line number. Refer to the Reference CEA-861D	

V_SYNC_LINE_BEF_1_1

Address = Base Address + 0x0000013C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	

V_SYNC_LINE_BEF_1	[4:0]	RW	v_sync_line_bef_1 [12:8] of 13 bits. Top field (or frame) V sync start line number. Refer to the Reference CEA-861D	
-------------------	-------	----	--	--

V_SYNC_LINE_AFT_2_0

Address = Base Address + 0x00000140, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_2	[7:0]	RW	v_sync_line_aft_2 [7:0] of 13 bits. Bottom field V sync end line number. Refer to the Reference CEA-861D	

V_SYNC_LINE_AFT_2_1

Address = Base Address + 0x00000144, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_2	[4:0]	RW	v_sync_line_aft_2 [12:8] of 13 bits. Bottom field V sync end line number. Refer to the Reference CEA-861D	

V_SYNC_LINE_AFT_1_0

Address = Base Address + 0x00000148, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_1	[7:0]	RW	v_sync_line_aft_1 [7:0] of 13 bits. Bottom field V sync start line number. Refer to the Reference CEA-861D	

V_SYNC_LINE_AFT_1_1

Address = Base Address + 0x0000014C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_1	[4:0]	RW	v_sync_line_aft_1 [12:8] of 13 bits. Bottom field V sync start line number. Refer to the Reference CEA-861D	

V_SYNC_LINE_AFT_PXL_2_0

Address = Base Address + 0x00000150, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_PXL_2	[7:0]	RW	v_sync_line_aft_pxl_2 [7:0] of 13 bits. Bottom field V sync end transition point. Refer to the Reference CEA-861D	

V_SYNC_LINE_AFT_PXL_2_1

Address = Base Address + 0x00000154, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_PXL_2	[4:0]	RW	v_sync_line_aft_pxl_2[12:8] of 13 bits. Bottom field V sync end transition point. Refer to the Reference CEA-861D	

V_SYNC_LINE_AFT_PXL_1_0

Address = Base Address + 0x00000158, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_PXL_1	[7:0]	RW	v_sync_line_aft_pxl_1[7:0] of 13 bits. Bottom field V sync start transition point. Refer to the Reference CEA-861D	

V_SYNC_LINE_AFT_PXL_1_1

Address = Base Address + 0x0000015C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_PXL_1	[4:0]	RW	v_sync_line_aft_pxl_1[12:8] of 13 bits. Bottom field V sync start transition point. Refer to the Reference CEA-861D	

V_BLANK_F2_0

Address = Base Address + 0x00000160, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_BLANK_F2	[7:0]	RW	v_blank_f2[7:0] of 13 bits. The start position of third field's active region. For 2D mode, This value is not used.	

V_BLANK_F2_1

Address = Base Address + 0x00000164, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_BLANK_F2	[4:0]	RW	v_blank_f2[12:8] of 13 bits. The start position of third field's active region. For 2D mode, This value is not used.	

V_BLANK_F3_0

Address = Base Address + 0x00000168, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_BLANK_F3	[7:0]	RW	v_blank_f3[7:0] of 13 bits. The end position of third field's active region. For 2D mode, This value is not used.	

V_BLANK_F3_1

Address = Base Address + 0x0000016C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_BLANK_F3	[4:0]	RW	v_blank_f3[12:8] of 13 bits. The end position of third field's active region. For 2D mode, This value is not used.	

V_BLANK_F4_0

Address = Base Address + 0x00000170, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_BLANK_F4	[7:0]	RW	v_blank_f4[7:0] of 13 bits. The start position of fourth field's active region. For 2D mode, This value is not used	

V_BLANK_F4_1

Address = Base Address + 0x00000174, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_BLANK_F4	[4:0]	RW	v_blank_f4[12:8] of 13 bits. The start position of fourth field's active region. For 2D mode, This value is not used.	

V_BLANK_F5_0

Address = Base Address + 0x00000178, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_BLANK_F5	[7:0]	RW	v_blank_f5[7:0] of 13 bits. The end position of fourth field's active region. For 2D mode, This value is not used.	

V_BLANK_F5_1

Address = Base Address + 0x0000017C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_BLANK_F5	[4:0]	RW	v_blank_f5[12:8] of 13 bits. The end position of fourth field's active region. For 2D mode, This value is not used.	

V_SYNC_LINE_AFT_3_0

Address = Base Address + 0x00000180, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_3	[7:0]	RW	v_sync_line_aft_3[7:0] of 13 bits. Third field V sync start line number.	

V_SYNC_LINE_AFT_3_1

Address = Base Address + 0x00000184, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_3	[4:0]	RW	v_sync_line_aft_3[12:8] of 13 bits. Third field V sync start line number.	

V_SYNC_LINE_AFT_4_0

Address = Base Address + 0x00000188, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_4	[7:0]	RW	v_sync_line_aft_4[7:0] of 13 bits. Third field V sync end line number.	

V_SYNC_LINE_AFT_4_1

Address = Base Address + 0x0000018C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_4	[4:0]	RW	v_sync_line_aft_4[12:8] of 13 bits. Third field V sync end line number.	

V_SYNC_LINE_AFT_5_0

Address = Base Address + 0x00000190, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_5	[7:0]	RW	v_sync_line_aft_5[7:0] of 13 bits. Fourth field V sync start line number.	

V_SYNC_LINE_AFT_5_1

Address = Base Address + 0x00000194, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_5	[4:0]	RW	v_sync_line_aft_5[12:8] of 13 bits. Fourth field V sync start line number.	

V_SYNC_LINE_AFT_6_0

Address = Base Address + 0x00000198, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_6	[7:0]	RW	v_sync_line_aft_6[7:0] of 13 bits. Fourth field V sync end line number.	

V_SYNC_LINE_AFT_6_1

Address = Base Address + 0x0000019C, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_6	[4:0]	RW	v_sync_line_aft_6[12:8] of 13 bits. Fourth field V sync end line number.	

V_SYNC_LINE_AFT_PXL_3_0

Address = Base Address + 0x000001A0, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_PXL_3	[7:0]	RW	v_sync_line_aft_pxl_3[7:0] of 13 bits. Third field V sync start transition point.	

V_SYNC_LINE_AFT_PXL_3_1

Address = Base Address + 0x000001A4, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_PXL_3	[4:0]	RW	v_sync_line_aft_pxl_3[12:8] of 13 bits. Third field V sync start transition point.	

V_SYNC_LINE_AFT_PXL_4_0

Address = Base Address + 0x000001A8, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_PXL_4	[7:0]	RW	v_sync_line_aft_pxl_4[7:0] of 13 bits. Third field V sync end transition point.	

V_SYNC_LINE_AFT_PXL_4_1

Address = Base Address + 0x000001AC, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_PXL_4	[4:0]	RW	v_sync_line_aft_pxl_4[12:8] of 13 bits. Third field V sync end transition point.	

V_SYNC_LINE_AFT_PXL_5_0

Address = Base Address + 0x000001B0, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
V_SYNC_LINE_AFT_PXL_5	[7:0]	RW	v_sync_line_aft_pxl_5[7:0] of 13 bits. Fourth field V sync start transition point.	

V_SYNC_LINE_AFT_PXL_5_1

Address = Base Address + 0x000001B4, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
V_SYNC_LINE_AFT_PXL_5	[4:0]	RW	v_sync_line_aft_pxl_5 [12:8] of 13 bits. Fourth field V sync start transition point.	

V_SYNC_LINE_AFT_PXL_6_0

Address = Base Address + 0x000001B8, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
v_sync_line_aft_pxl_6	[7:0]	RW	v_sync_line_aft_pxl_6 [7:0] of 13 bits. Fourth field V sync end transition point.	

V_SYNC_LINE_AFT_PXL_6_1

Address = Base Address + 0x000001BC, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
\$\$\$ 비어있음	\$\$\$	\$\$\$	v_sync_line_aft_pxl_6 [12:8] of 13 bits. Fourth field V sync end transition point.	

VACT_SPACE1_0

Address = Base Address + 0x000001C0, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
VACT_SPACE1	[7:0]	RW	vact_space1 [7:0] of 13 bits. first active space start line number	

VACT_SPACE1_1

Address = Base Address + 0x000001C4, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
VACT_SPACE1	[4:0]	RW	vact_space1 [12:8] of 13 bits. first active space start line number	

VACT_SPACE2_0

Address = Base Address + 0x000001C8, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
VACT_SPACE2	[7:0]	RW	vact_space2 [7:0] of 13 bits. first active space end line number	

VACT_SPACE2_1

Address = Base Address + 0x000001CC, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
VACT_SPACE2	[4:0]	RW	vact_space2 [12:8] of 13 bits. first active space end line number	

VACT_SPACE3_0

Address = Base Address + 0x000001D0, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
VACT_SPACE3	[7:0]	RW	vact_space3 [7:0] of 13 bits. second active space start line number	

VACT_SPACE3_1

Address = Base Address + 0x000001D4, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
VACT_SPACE3	[4:0]	RW	vact_space3 [12:8] of 13 bits. second active space start line number	

VACT_SPACE4_0

Address = Base Address + 0x000001D8, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
VACT_SPACE4	[7:0]	RW	vact_space4 [7:0] of 13 bits. second active space end line number	

VACT_SPACE4_1

Address = Base Address + 0x000001DC, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
VACT_SPACE4	[4:0]	RW	vact_space4 [12:8] of 13 bits. Third active space end line number	

VACT_SPACE5_0

Address = Base Address + 0x000001E0, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value

VACT_SPACE5	[7:0]	RW	vact_space5 [7:0] of 13 bits. Third active space start line number	
-------------	-------	----	---	--

VACT_SPACE5_1

Address = Base Address + 0x000001E4, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
VACT_SPACE5	[4:0]	RW	vact_space5 [12:8] of 13 bits. Third active space start line number	

VACT_SPACE6_0

Address = Base Address + 0x000001E8, Reset Value = 0xff

Name	Bit	Type	Description	Reset Value
VACT_SPACE6	[7:0]	RW	vact_space6 [7:0] of 13 bits. Third active space end line number	

VACT_SPACE6_1

Address = Base Address + 0x000001EC, Reset Value = 0x1f

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
VACT_SPACE6	[4:0]	RW	vact_space6 [12:8] of 13 bits. Third active space end line number	

GCP_CON

Address = Base Address + 0x00000200, Reset Value = 0x04

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	
ENABLE_1ST_VSYNC	[3]	RW	Enable this bit to transfer the GCP packet on the 1st VSYNC in a frame 0: Do not transfer GCP packet 1: Transfer GCP packet	
ENABLE_2ND_VSYNC	[2]	RW	For Interlace mode, Enable this bit to transfer the GCP packet on the 2nd VSYNC in a frame 0: Do not transfer GCP packet 1: Transfer GCP packet	
GCP_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync. GCP packet will be transmitted within 384 cycles after active vsync. After transferring first GCP packet, GCP_CON[0] is changed to 0.	

GCP_BYTE1

Address = Base Address + 0x00000210, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
GCP_BYTE1	[7:0]	RW	GCP packet's first data byte. It shall be either 0x10 (Clear AVMUTE) or 0x01 (Set AVMUTE). Refer to Table 5-17 of HDMI v1.3 specification	

GCP_BYTE2

Address = Base Address + 0x00000214, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
PP	[7:4]	RW	PP (Packing Phase), Read Only	
CD	[3:0]	RW	CD (Color Depth)	

GCP_BYTE3

Address = Base Address + 0x00000218, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
GCP_BYTE3	[0]	RW	Default State	

ASP_CON

Address = Base Address + 0x00000300, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
DST_DOUBLE	[7]	RW	DST double	
AUD_TYPE	[6:5]	RW	Packet type instead of audio type 00 : Audio Sample Packet 01 : One-bit audio packet 10 : HBR packet 11 : DST packet	
AUD_MODE	[4]	RW	Two channel or multi-channel mode selection This bit will be also used for layout bit in ASP header. 0 : 2 channel mode 1 : Multi channel mode. Set this bit to transmit HBR packets.	
SP_PRE	[3:0]	RW	Control sub-packet usage for multi channel mode only. When two-channel mode, this register value is not used.	

ASP_SP_FLAT

Address = Base Address + 0x00000304, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	

SP_FLAT	[3:0]	RW	The sp_flat/sample_invalid value in the ASP header. Refer to the HDMI specification v1.3 (5.3.4 and 5.3.9)	
----------------	-------	----	--	--

ASP_CHCFG0

Address = Base Address + 0x00000310, Reset Value = 0x08

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	
SPK0R_SEL	[5:3]	RW	Audio channel Selection for subpacket 0 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL	
SPK0L_SEL	[2:0]	RW	Audio channel Selection for subpacket 0 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL	

ASP_CHCFG1

Address = Base Address + 0x00000314, Reset Value = 0x1A

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	
SPK1R_SEL	[5:3]	RW	Audio channel Selection for subpacket 1 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL	
SPK1L_SEL	[2:0]	RW	Audio channel Selection for subpacket 1 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL	

ASP_CHCFG2

Address = Base Address + 0x00000318, Reset Value = 0x2C

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	
SPK2R_SEL	[5:3]	RW	Audio channel Selection for subpacket 2 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL	
SPK2L_SEL	[2:0]	RW	Audio channel Selection for subpacket 2 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL	

ASP_CHCFG3

Address = Base Address + 0x0000031C, Reset Value = 0x3E

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	

SPK3R_SEL	[5:3]	RW	Audio channel Selection for subpacket 3 right channel data in multi channel mode. 000 : i_pcm0L is used for sub packet 0 Left channel 001 : i_pcm0R is used for sub packet 0 Left channel 010 : i_pcm1L is used for sub packet 0 Left channel 011 : i_pcm1R is used for sub packet 0 Left channel 100 : i_pcm2L is used for sub packet 0 Left channel 101 : i_pcm2R is used for sub packet 0 Left channel 110 : i_pcm3L is used for sub packet 0 Left channel 111 : i_pcm3R is used for sub packet 0 Left channel	
SPK3L_SEL	[2:0]	RW	Audio channel Selection for subpacket 3 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL	

ACR_CON

Address = Base Address + 0x00000400, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
RSVD	[4:3]	R	Reserved	
ACR_TX_MODE	[2:0]	RW	000 : Do not Tx (Transfer) the ACR packet. 100 : Measured CTS mode. Make ACR packet with CTS value by counting TMDS clock for Fs x 128 / N duration. In this case, the 7 LSBs of N value (ACR_N register) should be all zero.	

ACR_MCTS0

Address = Base Address + 0x00000410, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
ACR_MCTS	[7:0]	R	ACR_MCTS [7:0] of 20 bits. This value is the TMDS clock cycles for N[19:7] number of audio sample inputs. It is only valid when measured CTS mode is set on ACR_CON register	

ACR_MCTS1

Address = Base Address + 0x00000414, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ACR_MCTS	[7:0]	R	ACR_MCTS [15:8] of 20 bits. This value is the TMDS clock cycles for N[19:7] number of audio sample inputs. It is only valid when measured CTS mode is set on ACR_CON register	

ACR_MCTS2

Address = Base Address + 0x00000418, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	

ACR_MCTS	[3:0]	R	ACR_MCTS [19:16] of 20 bits. This value is the TMDS clock cycles for N[19:7] number of audio sample inputs. It is only valid when measured CTS mode is set on ACR_CON register	
-----------------	-------	---	---	--

ACR_N0

Address = Base Address + 0x00000430, Reset Value = 0xE8

Name	Bit	Type	Description	Reset Value
ACR_N	[7:0]	RW	ACR_N [7:0] of 20 bits. The N value in ACR packet	

ACR_N1

Address = Base Address + 0x00000434, Reset Value = 0x03

Name	Bit	Type	Description	Reset Value
ACR_N	[7:0]	RW	ACR_N [15:8] of 20 bits. The N value in ACR packet	

ACR_N2

Address = Base Address + 0x00000438, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	
ACR_N	[3:0]	RW	ACR_N [19:16] of 20 bits. The N value in ACR packet	

ACP_CON

Address = Base Address + 0x00000500, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ACP_FR_RATE	[7:3]	RW	Transmit ACP packet once per every ACP_FR_RATE+1 frames (or fields).	
RSVD	[2]	RW	Reserved	
ACP_TX_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync with ACP_FR_RATE	

ACP_TYPE

Address = Base Address + 0x00000514, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ACP_TYPE	[7:0]	RW	ACP packet header. (HB1 of ACP packet header) See Table 5-18 in HDMI v1.3 specification	

ACP_DATAx

Address = Base Address + 0x00000520, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ACP_DATAx	[7:0]	RW	ACP packet body data. (PB0~PB16 of ACP packet body) See 9.3 in HDMI v1.3 specification	

ISRC_CON

Address = Base Address + 0x00000600, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC_FR_RATE	[7:3]	RW	Transmit ISRC1 (with ISRC2 or not) packet once per every ISRC_FR_RATE+1 frames (or fields).	
ISRC2_EN	[2]	RW	Transmit ISRC2 packet with ISRC1 packet	
ISRC_TX_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync with ISRC_FR_RATE	

ISRC1_HEADER1

Address = Base Address + 0x00000614, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC_CONT	[7]	RW	See table 5-20 in HDMI v1.3 specification	
ISRC_VALID	[6]	RW	See table 5-20 in HDMI v1.3 specification	
RSVD	[5:3]	RW	Reserved	
ISRC_STATUS	[2:0]	RW	See table 5-20 in HDMI v1.3 specification	

ISRC1_DATAx

Address = Base Address + 0x00000620, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC1_DATAx	[7:0]	RW	ISRC1 packet body data. (PB0~15 of ISRC1 packet body) See Table 5-21 in HDMI v1.3 specification.	

ISRC2_DATAx

Address = Base Address + 0x000006A0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ISRC2_DATAx	[7:0]	RW	ISRC2 packet body data. (PB0~15 of ISRC2 packet body) See Table 5-23 in HDMI v1.3 specification.	

AVI_CON

Address = Base Address + 0x00000700, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

RSVD	[7:2]	RW	Reserved	
AVI_TX_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	

AVI_HEADER0

Address = Base Address + 0x00000710, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_HEADER0	[7:0]	RW	HB0 byte of AVI packet header	

AVI_HEADER1

Address = Base Address + 0x00000714, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_HEADER1	[7:0]	RW	HB1 byte of AVI packet header	

AVI_HEADER2

Address = Base Address + 0x00000718, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_HEADER2	[7:0]	RW	HB2 byte of AVI packet header	

AVI_CHECK_SUM

Address = Base Address + 0x0000071C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_CHECK_SUM	[7:0]	RW	AVI InfoFrame checksum byte. (PB0 byte of AVI packet body)	

AVI_BYTEX

Address = Base Address + 0x00000720, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AVI_BYTEX	[7:0]	RW	AVI Infoframe packet data registers. (PB1~PB13 bytes of AVI packet body)	

AUI_CON

Address = Base Address + 0x00000800, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	

AUI_TX_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	
------------	-------	----	---	--

AUI_HEADER0

Address = Base Address + 0x00000810, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_HEADER0	[7:0]	RW	HB0 byte of AUI packet header	

AUI_HEADER1

Address = Base Address + 0x00000814, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_HEADER1	[7:0]	RW	HB1 byte of AUI packet header	

AUI_HEADER2

Address = Base Address + 0x00000818, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_HEADER2	[7:0]	RW	HB2 byte of AUI packet header	

AUI_CHECK_SUM

Address = Base Address + 0x0000081C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_CHECK_SUM	[7:0]	RW	AUI Infoframe checksum data. (PB0 byte of AUI packet body)	

AUI_BYTEX

Address = Base Address + 0x00000820, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AUI_BYTEX	[7:0]	RW	AUI Infoframe packet body. (PB1~PB12 bytes of AUI packet body)	

MPG_CON

Address = Base Address + 0x00000900, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
MPG_TX_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	

MPG_CHECK_SUM

Address = Base Address + 0x00000091C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
MPG_CHECK_SUM	[7:0]	RW	MPG infoframe checksum register (PB0 byte of MPG packet body)	

MPG_DATAx

Address = Base Address + 0x000000920, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
MPG_DTATX	[7:0]	RW	MPG Infoframe packet data. (PB1~PB5 bytes of MPG packet body)	

SPD_CON

Address = Base Address + 0x00000A00, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
SPD_TX_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	

SPD_HEADER0

Address = Base Address + 0x00000A10, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
SPD_HEADER0	[7:0]	RW	HB0 byte of SPD packet header	

SPD_HEADER1

Address = Base Address + 0x00000A14, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
SPD_HEADER1	[7:0]	RW	HB1 byte of SPD packet header	

SPD_HEADER2

Address = Base Address + 0x00000A18, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
SPD_HEADER2	[7:0]	RW	HB2 byte of SPD packet header	

SPD_DATAx

Address = Base Address + 0x00000A20, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

SPD_DATAx	[7:0]	RW	SPD packet data registers. (PB0~PB27 bytes)	
-----------	-------	----	---	--

GAMUT_CON

Address = Base Address + 0x00000B00, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
GAMUT_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	

GAMUT_HEADER0

Address = Base Address + 0x00000B10, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HBO	[7:0]	RW	HBO value in the table 5-30 in HDMI 1.3 specification	

GAMUT_HEADER1

Address = Base Address + 0x00000B14, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
NEXT_FIELD	[7]	RW	Set to indicate that the GBD carried in this packet will be effective on the next video field.	
GBD_PROFILE	[6:4]	RW	Transmission profile number (We only support profile 0)	
AFFECTED_GAMUT_SEQ_NUM	[3:0]	RW	Indicates which video fields are relevant for this metadata	

GAMUT_HEADER2

Address = Base Address + 0x00000B18, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
NO_CRNT_GBD	[7]	RW	Set to indicate that there is no gamut metadata available for the currently transmitted video	
RESERVED	[6]	RW	Reserved	
PACKET_SEQ	[5:4]	RW	Indicates whether this packet is the only, the first, an intermediate or the last packet in a Gamut metadata packet sequence	
CURRENT_GAMUT_SEQ_NUM	[3:0]	RW	Indicates the gamut number of the currently transmitted video stream	

GAMUT_METADATAx

Address = Base Address + 0x00000B20, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
------	-----	------	-------------	-------------

GAMUT_METADATAAX	[7:0]	RW	Gamut Metadata Packet body for P0 transmission profile	
------------------	-------	----	--	--

VSI_CON

Address = Base Address + 0x00000C00, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
VSI_TX_CON	[1:0]	RW	00 : Do not transmit 01 : Transmit once 1x : Transmit every vsync	

VSI_HEADER0

Address = Base Address + 0x00000C10, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_HEADER0	[7:0]	RW	HB0 byte of VSI packet header	

VSI_HEADER1

Address = Base Address + 0x00000C14, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_HEADER1	[7:0]	RW	HB1 byte of VSI packet header	

VSI_HEADER2

Address = Base Address + 0x00000C18, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_HEADER2	[7:0]	RW	HB2 byte of VSI packet header	

VSI_DATAx

Address = Base Address + 0x00000C20, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VSI_DATAx	[7:0]	RW	VSI packet data registers. (PB0~PB27 bytes)	

DC_CONTROL

Address = Base Address + 0x00000D00, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	

DEEP_COLOR_MODE	[1:0]	RW	00: 8 bits /pixel 01: 10bits /pixel 10: 12 bits / pixel 11: Not Used	
------------------------	-------	----	---	--

VIDEO_PATTERN_GEN

Address = Base Address + 0x00000D04, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
EXT_VIDEO_EN	[1]	RW	0: Ext Off 1: Ext En	
VIDEO_PATTERN_ENABLE	[0]	RW	0: Disable 1: Use Internally generated video pattern	

An_Seed_Sel

Address = Base Address + 0x00000E48, Reset Value = 0xFF

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
AN_SEED_SEL	[0]	RW	0: use An_Seed_0~3 registers as a seed. 1: use input R/G/B as a seed.	

An_Seed_0

Address = Base Address + 0x00000E58, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AN_SEED	[7:0]	RW	[23:16] of An seed value	

An_Seed_1

Address = Base Address + 0x00000E5C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	
AN_SEED	[3:0]	RW	[15:12] of An seed value	

An_Seed_2

Address = Base Address + 0x00000E60, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AN_SEED	[7:0]	RW	[11:4] of An seed value	

An_Seed_3

Address = Base Address + 0x00000E64, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	
AN_SEED	[3:0]	RW	[3:0] of An seed value	

HDCP_SHA1_x

Address = Base Address + 0x00007000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_SHA1_X	[7:0]	RW	<p>An 160-bit HDCP repeater's SHA-1 value. Least significant byte first. For example,</p> <ul style="list-style-type: none"> • SHA-1 Value[7:0] <= HDCP_SHA1_00[7:0] • SHA-1 Value[159:152] <= HDCP_SHA1_19[7:0] <p>These registers are readable but they are not modified by HDCP HW.</p> <p>* Note: Writing to HDCP_SHA1_00~19 register (any byte), regardless of the write value, triggers the SHA1 module to start the calculation.</p> <p>Do not write these register for R/W testing. Write only when calculating SHA1 value.</p>	

HDCP_KSV_LIST_x

Address = Base Address + 0x00007050, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_KSV_LIST_X	[7:0]	RW	<p>One 40-bit KSV value of the HDCP repeater's KSV list. These registers are readable.</p> <p>Least significant byte first. For example,</p> <ul style="list-style-type: none"> • KSV value [7:0] <= HDCP_KSV_LIST_0[7:0] • KSV value [39:32] <= HDCP_KSV_LIST_4[7:0] 	

HDCP_KSV_LIST_CON

Address = Base Address + 0x00007064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	
HDCP_KSV_WRITE_DONE	[3]	RW	<p>After writing KSV data into HDCP_KSV_LIST_X registers and then writing the value "1" to this register, HW processes the written KSV value and clears this bit to "0"</p> <p>0: Not yet written 1: Written</p>	
HDCP_KSV_LIST_EMPTY	[2]	RW	<p>If the number of KSV list is zero, set this value to make SHA-1 module to start to calculate without KSV list.</p> <p>0: Not empty 1: Empty</p>	
HDCP_KSV_END	[1]	RW	<p>It is used to indicate that current KSV value in HDCP_KSV_LIST_X registers is the last one.</p> <p>0: Not End 1: End</p>	

HDCP_KSV_READ	[0]	RW	After writing KSV data into HDCP_KSV_LIST_X registers HD-CP SHA-1 module keeps that KSV value into internal buffer and set this flag into 1 to notify that it has been read. After checking that it is set to 1, SW clears to 0 at the same time when writing the HDCP_KSV_WRITE_DONE bit for next KSV list value. 0: Not Read 1: Read	
---------------	-----	----	--	--

HDCP_SHA_RESULT

Address = Base Address + 0x00007070, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
HDCP_SHA_VALID_READY	[1]	RW	Indicates that the SHA comparison has been done by the HW. Must be cleared by SW by writing 0 0: Not ready 1: Ready	
HDCP_SHA_VALID	[0]	RW	Indicates that the SHA-1 comparison succeeds. Must be cleared by SW by writing 0 0: Not Valid 1: Valid	

HDCP_CTRL1

Address = Base Address + 0x00007080, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	
RSVD	[3]	RW	Reserved	
IMEOUT	[2]	RW	Set when Rx is repeater and its KSV list is not ready until 5 sec waiting. 0: Not timeout 1: Timeout(KSV Ready bit in the HDCP_BCAPS register is not high until 5 sec) and re-start the 1st authentication.	
CP_DESIRED	[1]	RW	HDCP enable 0: Not Desired 1: Desired	
RSVD	[0]	RW	Reserved	

HDCP_CTRL2

Address = Base Address + 0x00007084, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
REVOCATION_SET	[0]	RW	KSV list is on the revocation list & Fail the 2nd authentication. 0: Revocation Not set 1: Revocation Set	

HDCP_CHECK_RESULT

Address = Base Address + 0x00007090, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
RI_MATCH_RESULT	[1:0]	RW	Write the result of comparison between Ri of Rx and Tx as the following values. (Ri : Tx, R'i : Rx) Must be cleared by SW after setting 10 or 11 before next Ri interrupt occurs. 0x : don't care 10 : Ri < R'i 11 : Ri = R'i	

HDCP_BKSV_X

Address = Base Address + 0x000070A0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_BKSV_X	[7:0]	RW	Key selection vector (KSV) value from receiver. HDCP_BKSV [7:0] <= HDCP_BKSV_0 [7:0] HDCP_BKSV [15:8] <= HDCP_BKSV_1 [7:0] HDCP_BKSV [23:16] <= HDCP_BKSV_2 [7:0] HDCP_BKSV[31:24]<=HDCP_BKSV_3[7:0]HDCP_BKSV[39:32]<=HDCP_BKSV_4[7:0]	

HDCP_AKSV_X

Address = Base Address + 0x000070C0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_AKSV_X	[7:0]	RW	KSV value of transmitter HDCP_AKSV [7:0] <= HDCP_AKSV_0 [7:0] HDCP_AKSV [15:8] <= HDCP_AKSV_1 [7:0] HDCP_AKSV [23:16] <= HDCP_AKSV_2 [7:0] HDCP_AKSV[31:24]<=HDCP_AKSV_3[7:0]HDCP_AKSV[39:32]<=HDCP_AKSV_4[7:0]	

HDCP_An_X

Address = Base Address + 0x000070E0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_AN_X	[7:0]	RW	64-bit Random number generated by Tx (An) HDCP_An [7:0] <= HDCP_An_0_0 [7:0] HDCP_An [15:8] <= HDCP_An_1 [7:0] HDCP_An [23:16] <= HDCP_An_2 [7:0] HDCP_An [31:24] <= HDCP_An_3 [7:0] HDCP_An [39:32] <= HDCP_An_4 [7:0] HDCP_An [47:40] <= HDCP_An_5 [7:0] HDCP_An [55:48] <= HDCP_An_6 [7:0] HDCP_An [63:56] <= HDCP_An_7 [7:0]	

HDCP_BCAPS

Address = Base Address + 0x00007100, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	
REPEATER	[6]	RW	The receiver supports downstream connections 0: Not Repeater 1: Repeater	

READY	[5]	RW	KSV FIFO, SHA-1 calculation ready 0: Not Ready 1: Ready	
FAST	[4]	RW	The receiver devices supports 400KHz transfer 0: Not Fast 1: Fast	
RESERVED	[3:2]	RW	Must be 0's	
V1P1_FEATURES	[1]	RW	Supports EESS, Advance cipher, Enhanced link verification 0: un-set 1: set	
FAST_REAUTHENTICATION	[0]	RW	ALL HDMI receiver should be capable of reauthentication 0: un-set 1: set	

HDCP_BSTATUS_0

Address = Base Address + 0x00007110, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
MAX_DEVS_EXCEEDED	[7]	RW	Topology error indicator 0: No Error 1: Error	
DEVICE_COUNT	[6:0]	RW	Total number of attached downstream devices	

HDCP_BSTATUS_1

Address = Base Address + 0x00007114, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	RW	Reserved	
HDMI_MODE	[4]	RW	HDMI mode indication. If set, HDCP is in HDMI mode.	
MAX CASCADE EXCEEDED	[3]	RW	Topology error	
DEPTH	[2:0]	RW	Cascade depth	

HDCP_Ri_0

Address = Base Address + 0x00007140, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HDCP_RI	[7:0]	R	HDCP Ri value of the transmitter. HDCP_Ri [7:0] of 16 bits.	

HDCP_Ri_1

Address = Base Address + 0x00007144, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

HDCP_RI	[7:0]	R	HDCP Ri value of the transmitter. HDCP_Ri [15:8] of 16 bits.	
---------	-------	---	--	--

HDCP_I2C_INT

Address = Base Address + 0x00007180, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HDCP_I2C_INT	[0]	RW	HDCP I2C Interrupt status. Active high. It indicates the start of I2C transaction when it is set. After active, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	

HDCP_AN_INT

Address = Base Address + 0x00007190, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HDCP_AN_INT	[0]	RW	HDCP An Interrupt status. Active high. If An value is available, it is set. After active, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	

HDCP_WATCGDOG_INT

Address = Base Address + 0x000071A0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HDCP_WATCHDOG_INT	[0]	RW	HDCP Watchdog Interrupt status. Active high. If Repeater bit value is set after 1st authentication success, it is set. After active, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	

HDCP_RI_INT

Address = Base Address + 0x000071B0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
HDCP_RI_INT	[0]	RW	If Ri value is updated internally (at every 128 video frames), it is set to high. After set, it should be cleared by S/W by writing 0. 0: Not Occurred 1: Occurred	

HDCP_Ri_Compare_0

Address = Base Address + 0x000071D0, Reset Value = 0x80

Name	Bit	Type	Description	Reset Value
ENABLE	[7]	RW	Enable the interrupt for this frame number index	
FRAME_NUMBER_INDEX	[6:0]	RW	When the Frame count reaches this "Frame number index", an Ri Link integrity check interrupt occurs	

HDCP_Ri_Compare_1

Address = Base Address + 0x000071D4, Reset Value = 0x7F

Name	Bit	Type	Description	Reset Value
ENABLE	[7]	RW	Enable the interrupt for this frame number index	
FRAME_NUMBER_INDEX	[6:0]	RW	When the Frame count reaches this "Frame number index", an Ri Link integrity check interrupt occurs	

HDCP_Frame_Count

Address = Base Address + 0x000071E0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	
FRAME_COUNT	[6:0]	R	Current value of the frame count index in the hardware	

RGB_ROUND_EN

Address = Base Address + 0x0000D500, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	R	Reserved	
RGB_ROUND_EN	[0]	RW	RGB Rounding enable	

VACT_SPACE_R_0

Address = Base Address + 0x0000D504, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VACT_SPACE_R	[7:0]	RW	vact_space_r [7:0] of 12 bits. Constant pixel color in vact space.	

VACT_SPACE_R_1

Address = Base Address + 0x0000D508, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	

VACT_SPACE_R	[3:0]	RW	vact_space_r[11:8] of 12 bits. Constant pixel color in vact space.	
--------------	-------	----	--	--

VACT_SPACE_G_0

Address = Base Address + 0x0000D50C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VACT_SPACE_G	[7:0]	RW	vact_space_g[7:0] of 12 bits. Constant pixel color in vact space.	

VACT_SPACE_G_1

Address = Base Address + 0x0000D510, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	
VACT_SPACE_G	[3:0]	RW	vact_space_g[11:8] of 12 bits. Constant pixel color in vact space.	

VACT_SPACE_B_0

Address = Base Address + 0x0000D514, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
VACT_SPACE_B	[7:0]	RW	vact_space_b[7:0] of 12 bits. Constant pixel color in vact space.	

VACT_SPACE_B_1

Address = Base Address + 0x0000D518, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	
VACT_SPACE_B	[3:0]	RW	vact_space_b[11:8] of 12 bits. Constant pixel color in vact space.	

BLUE_SCREEN_R_0

Address = Base Address + 0x0000D520, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BLUE_SCREEN_R	[7:0]	RW	blue_screen_r[7:0] of 12 bits.	

BLUE_SCREEN_R_1

Address = Base Address + 0x0000D524, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

RSVD	[7:4]	R	Reserved	
BLUE_SCREEN_R	[3:0]	RW	blue_screen_r [11:8] of 12 bits.	

BLUE_SCREEN_G_0

Address = Base Address + 0x0000D528, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BLUE_SCREEN_G	[7:0]	RW	blue_screen_g [7:0] of 12 bits.	

BLUE_SCREEN_G_1

Address = Base Address + 0x0000D52C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	
BLUE_SCREEN_G	[3:0]	RW	blue_screen_g [11:8] of 12 bits.	

BLUE_SCREEN_B_0

Address = Base Address + 0x0000D530, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BLUE_SCREEN_B	[7:0]	RW	blue_screen_b [7:0] of 12 bits.	

BLUE_SCREEN_B_1

Address = Base Address + 0x0000D534, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	
BLUE_SCREEN_B	[3:0]	RW	blue_screen_b [11:8] of 12 bits.	

37.3.3 AES Registers

Register Map Summary (Base Address 0xC022_0000)

Register	Offset	Description	Reset Value
AES_START	0x00000000	AES_START	0x00
AES_DATA_SIZE_L	0x00000020	AES_DATA_SIZE_L	0x20
AES_DATA_SIZE_H	0x00000024	AES_DATA_SIZE_H	0x01
AES_DATA	0x00000040	AES_DATA	0x00

AES_START

Address = Base Address + 0x00000000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
AES_START	[0]	RW	AES Start signal If specified amount of data is decrypted and written in memory, then AES start signal goes to 0. 0 : AES does not decrypt data. (AES decryption completed) 1 : AES starts to decrypt data from memory.	

AES_DATA_SIZE_L

Address = Base Address + 0x00000020, Reset Value = 0x20

Name	Bit	Type	Description	Reset Value
AES_DATA_SIZE_L	[7:0]	RW	AES data size (in bytes) to decrypt If the data size is not the multiple of 128 bits, zeros should be padded 128bit align Maximum number is 120h(internal memory size limits the maximum data size Default value : 288 bytes	

AES_DATA_SIZE_H

Address = Base Address + 0x00000024, Reset Value = 0x01

Name	Bit	Type	Description	Reset Value
AES_DATA_SIZE_H	[7:0]	RW	AES data size (in bytes) to decrypt If the data size is not the multiple of 128 bits, zeros should be padded 128bit align Maximum number is 120h(internal memory size limits the maximum data size Default value : 288 bytes	

AES_DATA

Address = Base Address + 0x00000040, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
AES_DATA	[7:0]	RW	Write buffer to store AES-encrypted data in memory before starting decryption Memory address is automatically increased. Zeros should be padded for 128 bit align.	

37.3.4 SPDIF Registers

Register Map Summary (Base Address 0xC023_0000)

Register	Offset	Description	Reset Value
SPDIFIN_CLK_CTRL	0x00000000	SPDIFIN Clock Control Register	0x02
SPDIFIN_OP_CTRL	0x00000004	SPDIFIN Operation Control Register 1	0x00
SPDIFIN_IRQ_MASK	0x00000008	SPDIFIN Interrupt Request Mask Register	0x00
SPDIFIN_IRQ_STATUS	0x0000000C	SPDIFIN Interrupt Request Status Register	0x00

SPDIFIN_CONFIG_1	0x00000010	SPDIFIN Configuration Register 1	0x02
SPDIFIN_CONFIG_2	0x00000014	SPDIFIN Configuration Register 2	0x00
SPDIFIN_USER_VALUE_1	0x00000020	SPDIFIN User Value Register 1	0x00
SPDIFIN_USER_VALUE_2	0x00000024	SPDIFIN User Value Register 2	0x00
SPDIFIN_USER_VALUE_3	0x00000028	SPDIFIN User Value Register 3	0x00
SPDIFIN_USER_VALUE_4	0x0000002C	SPDIFIN User Value Register 4	0x00
SPDIFIN_CH_STATUS_0_1	0x00000030	SPDIFIN Channel Status Register 0-1	0x00
SPDIFIN_CH_STATUS_0_2	0x00000034	SPDIFIN Channel Status Register 0-2	0x00
SPDIFIN_CH_STATUS_0_3	0x00000038	SPDIFIN Channel Status Register 0-3	0x00
SPDIFIN_CH_STATUS_0_4	0x0000003C	SPDIFIN Channel Status Register 0-4	0x00
SPDIFIN_CH_STATUS_1	0x00000040	SPDIFIN Channel Status Register 1	0x00
SPDIFIN_FRAME_PERIOD_1	0x00000048	SPDIFIN Frame Period Register 1	0x00
SPDIFIN_FRAME_PERIOD_2	0x0000004C	SPDIFIN Frame Period Register 2	0x00
SPDIFIN_PC_INFO_1	0x00000050	SPDIFIN Pc Info Register 1	0x00
SPDIFIN_PC_INFO_2	0x00000054	SPDIFIN Pc Info Register 2	0x00
SPDIFIN_PD_INFO_1	0x00000058	SPDIFIN Pd Info Register 1	0x00
SPDIFIN_PD_INFO_2	0x0000005C	SPDIFIN Pd Info Register 2	0x00
SPDIFIN_DATA_BUF_0_1	0x00000060	SPDIFIN Data Buffer Register 0_1	0x00
SPDIFIN_DATA_BUF_0_2	0x00000064	SPDIFIN Data Buffer Register 0_2	0x00
SPDIFIN_DATA_BUF_0_3	0x00000068	SPDIFIN Data Buffer Register 0_3	0x00
SPDIFIN_USER_BUF_0	0x0000006C	SPDIFIN User Buffer Register 0	0x00
SPDIFIN_DATA_BUF_1_1	0x00000070	SPDIFIN Data Buffer Register 1_1	0x00
SPDIFIN_DATA_BUF_1_2	0x00000074	SPDIFIN Data Buffer Register 1_2	0x00
SPDIFIN_DATA_BUF_1_3	0x00000078	SPDIFIN Data Buffer Register 1_3	0x00
SPDIFIN_USER_BUF_1	0x0000007C	SPDIFIN User Buffer Register 1	0x00

SPDIFIN_CLK_CTRL

Address = Base Address + 0x00000000, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
READY_CLK_DOWN	[1]	RW	0: clock is enabled 1: ready for disabling clock (default)	

POWER_ON	[0]	RW	<p>0: clock will be disabled (default) 1: clock will be activated</p> <p>If this bit is reset, SPDIFIN stops checking the input signal just before next subframe of SPDIF signal format and wait the <i>acknowledge</i> signal from HDMI for unresolved previous <i>request</i> toward HDMI. Then asserts <i>ready_clk_down</i> as HIGH.</p> <p>To initialize internal states, you have to assert S/W reset, i.e. SPDIFIN_OP_CTRL. op_ctrl=00b right after activating clock again.</p>	
-----------------	-----	----	--	--

SPDIFIN_OP_CTRL

Address = Base Address + 0x00000004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
OP_CTRL	[1:0]	RW	<p>00b : software reset 01b : status checking mode (run) 11b : status checking + HDMI operation mode (run with HDMI) Others : undefined, do not use</p> <p>00b : During a software reset, all state machines are set to the idle or init state and all internal registers are set to their initial values. Interrupt status registers are cleared, all other registers that are writable by the system processor keep their values.</p> <p>01b : This command should be asserted after SPDIFIN_CLK_CTRL.power_on is set. SPDIFIN starts clock recovery. When recovery is done, SPDIFIN begins detecting preambles of SPDIF signal format and stream data header, abnormal time signal input, abnormal signal input, and also reports these status via interrupts in SPDIFIN_IRQ_STATUS.</p> <p>11b : "01b" case operations + checking internal buffer overflow + write received data, which can be either audio sample word of PCM or payload of stream. Data will be transferred via HDMI.</p> <ul style="list-style-type: none"> You should assert op_ctrl=11b after SPDIFIN_IRQ_STATUS.ch_status_recovered_ir was asserted at least once for linear PCM data. Or you should assert op_ctrl=11b after SPDIFIN_IRQ_STATUS.stream_header_detected_ir was asserted at least once for non-linear PCM stream data. 	

SPDIFIN_IRQ_MASK

0: interrupt generation is disabled. 1: interrupt generation is enabled

Address = Base Address + 0x00000008, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BUF_OVERFLOW_IR_EN	[7]	RW	mask bit for Interrupt 7	
RSVD	[6]	RW	Reserved	
RSVD	[5]	RW	Reserved	
STREAM_HDR_DET_IR_EN	[4]	RW	mask bit for Interrupt 4 (stream header detected interrupt enable)	
STREAM_HDR_NOT_DET_IR_EN	[3]	RW	mask bit for Interrupt 3 (stream header not detected interrupt enable)	
WRONG_PREAMBLE_IR_EN	[2]	RW	mask bit for Interrupt 2	
CH_STATUS_RECOVERED_IR_EN	[1]	RW	mask bit for Interrupt 1	
WRONG_SIGNAL_IR_EN	[0]	RW	mask bit for Interrupt 0	

SPDIFIN_IRQ_STATUS

For every bit the following holds: Reading returns interrupt request status. Writing 0 has no effect. Writing 1 clears the interrupt request. 1) Detection of stream header Wait for matching of Pa, Pb; 0xF872, 0x4E1F respectively Wait for the repetition time (From decoded Pc value or from user-set Pc in SPDIFIN_USER_VALUE.repetition_time_manual according to SPDIFIN_CONFIG.PcPd_value_mode) Check for matching of Pa, Pb on right time.

Address = Base Address + 0x0000000C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BUF_OVERFLOW_IR	[7]	RW	<p>0: no interrupt 1: internal buffer overflow</p> <p>SPDIFIN internal buffer(s) (SPDIFIN_DATA_BUF_x) was over-flowed because HDMI did not transfer the data in the buffer(s) to memory in time.</p> <ul style="list-style-type: none"> This interrupt will be asserted only if SPDIFIN_OP_CTRL.op_ctrl was set as "011". If user does not handle this interrupt, SPDIFIN will over-write next subframe data to the internal data buffer (SPDIFIN_DATA_BUF_x) and continue data transfer via HDMI. 	
RSVD	[6]	RW	Reserved	
RSVD	[5]	RW	Reserved	
STREAM_HDR_DET_IR	[4]	RW	<p>0: no interrupt 1: stream data header (Pa-Pd) detected</p> <ul style="list-style-type: none"> This interrupt will be asserted when SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. Cases for interrupt <p>case1 : Initially after power_on</p> <p>case2 : Next stream header at right time when receiving stream data with SPDIFIN_CONFIG.data_type set as <i>stream mode</i>.</p> <p>case3: Initially detected stream header when receiving stream data with</p>	
STREAM_HDR_NOT_DET_IR	[3]	RW	<p>0: no interrupt 1: stream data header not detected for 4096 repetition time</p> <ul style="list-style-type: none"> This interrupt will be asserted when SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. Cases for interrupt <p>case1 : Initially after power_on</p> <p>case2 : SPDIFIN was receiving stream but could not find next stream header for 4096 repetition time since previous stream header</p> <p>case3 : Could not find stream header for 4096 repetition time since previous reset of repetition time counter after previous interrupt of <i>stream_header_not_detected_ir</i>.</p>	
WRONG_PREAMBLE_IR	[2]	RW	<p>0: no interrupt 1: preamble was detected but there is a problem with detected time</p> <ul style="list-style-type: none"> This interrupt will be asserted when SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. Meaningless until ch_status_recovered_ir is asserted initially after SPDIFIN_OP_CTRL.op_ctrl=01b Cases for interrupt <p>case1: preamble was detected in the middle of a subframe audio sample word time</p> <p>case2: next preamble was not detected at exact time after a sub-frame duration</p> <p>case3: it was time for preamble B(or M or W) to be detected but other preamble was detected at that time</p>	

CH_STATUS_RECOVERED_IR	[1]	RW	<p>0: no interrupt 1: recovered channel status Detected preamble of 2 consecutive B-preamble thus recovered 192 bit wide channel status.</p> <ul style="list-style-type: none"> Only supports consumer mode, so just 36bits will be reconstructed. If a user wants to see the channel status bits through SPDIFIN_CH_STATUS_x, you'd better read two consecutive <i>ch_status_recovered_ir</i> and read that register each time; if these two channel status value are same, you can rely on that value. 	
WRONG_SIGNAL_IR	[0]	RW	<p>0: no interrupt 1: clock recovery fail Can not recover clock from input because of tolerable range violation(unlock) or because of no signal from outside or because of non-biphase in non-preamble duration</p> <ul style="list-style-type: none"> Meaningless until <i>ch_status_recovered_ir</i> is asserted initially after SPDIFIN_OP_CTRL.op_ctrl=01b 	

SPDIFIN_CONFIG_1

Address = Base Address + 0x00000010, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	
NOISE_FILTER_SAMPLES	[6]	RW	<p>0: filtering with 3 consecutive samples 1: filtering with 2 consecutive samples Noise filtering is done for over-sampled SPDIF input signal. This operation will be done as follows. If <i>noise_filter_samples</i> is 0, 3 consecutive over-sampled signal will be regarded as a high or low only when those 3 samples are all high or low respectively. If 1 or 2 samples are low or high respectively for 3 over-sample duration, that noise filtered signal will keep previous value. If <i>noise_filter_samples</i> is 1, 2 consecutive over-sampled signal will be regarded as a high or low only when those 2 samples are all high or low respectively. This setting can be used for reduced over-sampling ratio; recommended over-sampling ratio is 10 (see also <i>clk_divisor</i>)</p>	
RSVD	[5]	RW	Reserved (Must be 0)	
PCPD_VALUE_MODE	[4]	RW	<p>0: automatically set 1: manually set If 0 for automatic setting, P_c, P_d values will be chosen by value of P_c, P_d from decoded stream header reported as in SPDIFIN_Px_INFO. If user sets this register, the receiver will use SPDIFIN_USER_VALUE[31:16], SPDIFIN_USER_VALUE[15:4] value as P_c, P_d respectively instead of decoded data from stream header as reported in SPDIFIN_Px_INFO. (cf) Burst payload length, whether it is automatically set or manually set, will affect the data size to be written in memory via HDMI by dumping the full sub-frame for the last bit for burst payload length. For example, if burst payload length is 257bit, i.e. (16sub-frame * 16bit + 1bit), then HDMI will write data in 17 consecutive sub-frames.</p>	

WORD_LENGTH_VALUE_MODE	[3]	RW	0: automatically set 1: manually set If 0 for automatic setting, word length value will be chosen by value of channel status from decoded SPDIF format as reported in SPDIFIN_CH_STATUS_1.word_length. If user sets this register, the receiver will use SPDIFIN_USER_VALUE[3:0] value as word length instead of decoded data from channel status as reported in SPDIFIN_CH_STATUS_1.word_length.	
U_V_C_P_REPORT	[2]	RW	0: neglects user_bit, validity_bit, channel status parity_bit of SPDIF format. 1: reports user_bit, validity_bit, channel status parity_bit of SPDIF format Report will be via HDMI for each subframe. Valid only if SPDIFIN_CONFIG.data_align is set for 32bit mode; see also SPDIFIN_DATA_BUF_X.	
RSVD	[1]	RW	Reserved (Must be 1)	
DATA_ALIGN	[0]	RW	0: 16bit mode 1: 32bit mode 16bit: only takes 16bits from MSB in a subframe of SPDIF format then concatenates two consecutive 16bit data in one 32bit register of SPDIFIN_DATA_BUF_X. 32bit: data from one subframe with zero padding to MSB part. (ex: 0x00ffff for 24bit data) When stream mode, you should set word_length_value_mode as 1 and set SPDIFIN_USER_VALUE.word_length_manual as 0b000. These two modes will be applied to both modes of SPDIFIN_CONFIG.data_type, i.e. PCM or stream; see also SPDIFIN_DATA_BUF_X.	

SPDIFIN_CONFIG_2

Address = Base Address + 0x00000014, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	RW	Reserved	
CLK_DIVISOR	[3:0]	RW	SPDIFIN_internal_clock = system_clock / (clk_divisor + 1) (SPDIFIN_internal_clock ≤ 135 Mhz) SPDIFIN over-samples the SPDIF input signal with internally made clock which is divided from system clock. Recommended over-sampling ratio is 8~10, thus following calculation holds. Recommended SPDIFIN_internal_clock = Sampling Frequency of SPDIF Input Signal 64 bits 10 times-over-sampling (ex) 48 kHz 64 bits 10 times-over-sampling = 31 Mhz	

SPDIFIN_USER_VALUE_1

Address = Base Address + 0x00000020, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
REPETITION_TIME_MANUAL_LOW	[7:4]	R	Repetition time[3:0] Repetition_time_manual register 12bits value. This register is low 4bits. Will be used for counting one block of stream data; valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: frames (1 frame = 2 subframes) of SPDIF format) The value should be (actual repetition time - 1). For example, if you want to use 1536 as manually set repetition time, you should write 1535.	

WORD_LENGTH_MANUAL	[3:0]	R	<p>Word length</p> <p>Will be used as size for transferring data to memory via HDMI; valid only when SPDIFIN_CONFIG.word_length_value_mode is set for manual mode; see also SPDIFIN_DATA_BUFA_x [0] is 1 [0] is 0 [3:1]</p> <p>101: 24 bits 20 bits 001: 23 bits 19 bits 010: 22 bits 18 bits 011: 21 bits 17 bits 100: 20 bits 16 bits</p>	
--------------------	-------	---	---	--

SPDIFIN_USER_VALUE_2

Address = Base Address + 0x00000024, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
REPETITION_TIME_MANUAL_HIGH	[7:0]	R	<p>Repetition time[11:4]</p> <p>Repetition_time_manual register 12bits value. This register is high 8bits.</p> <p>Will be used for counting one block of stream data; valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode.</p> <p>(Unit: frames (1 frame = 2 subframes) of SPDIF format) The value should be (actual repetition time - 1). For example, if you want to use 1536 as manually set repetition time, you should write 1535.</p>	

SPDIFIN_USER_VALUE_3

Address = Base Address + 0x00000028, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BURST_PAYLOAD_LEN_MANUAL_LO_W	[7:0]	R	<p>Burst_payload_length_manual[7:0]</p> <p>Burst_payload_length register is 16bits value. This register is low 8bits</p> <p>Valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode.</p> <p>(Unit: bits)</p>	

SPDIFIN_USER_VALUE_4

Address = Base Address + 0x0000002C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BURST_PAYLOAD_LEN_MANUAL_HIGH	[7:0]	R	<p>Burst_payload_length_manual[15:8] Burst_payload_length register is 16bits value. this register is high 8bits</p> <p>Valid only when SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode.</p> <p>(Unit: bits)</p>	

SPDIFIN_CH_STATUS_0_1

This register will be updated every 192 frames(1 block) of SPDIF format. SPDIFIN_CH_STATUS_0_1 [7:0] is matched internal register SPDIFIN_CH_STATUS_0 [7:0].

Address = Base Address + 0x00000030, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CHANNEL_STATUS_MODE	[7:6]	R	00: mode 0 others: reserved	
EMPHASIS	[5:3]	R	000: emphasis not indicated 100: emphasis - CD type	
COPYRIGHT_ASSERTION	[2]	R	0: copyright 1: no copyright	
AUDIO_SAMPLE_WORD	[1]	R	0: linear PCM 1: non-linear PCM	
CHANNEL_STATUS_BLOCK	[0]	R	0: consumer format 1: professional format	

SPDIFIN_CH_STATUS_0_2

Address = Base Address + 0x00000034, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CATEGORY_CODE	[7:0]	R	equipment type : [8:15] CD player: 1000_0000 DAT player: 1100_000L DCC player: 1100_001L Mini disc: 1001_001L (L : information about generation status of the material)	

SPDIFIN_CH_STATUS_0_3

Address = Base Address + 0x00000038, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CHANNEL_NUMBER	[7:4]	R	Channel Number (bit 20 is LSB)	
SOURCE_NUMBER	[3:0]	R	Source Number (bit 16 is LSB)	

SPDIFIN_CH_STATUS_0_4

Address = Base Address + 0x0000003C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	R	Reserved	
CLOCK_ACCURACY	[5:4]	R	Clock accuracy 00: level II, ±1000ppm 01: level I, ±50ppm 10: level III, variable pitch shifted	
SAMPLING_FREQUENCY	[3:0]	R	Sampling Frequency 0100: 22.05kHz 0000: 44.1 kHz 1000: 88.2kHz 1100: 176.4kHz 0110: 24kHz 0010: 48 kHz 1010: 96kHz 1110: 192kHz	

SPDIFIN_CH_STATUS_1

Address = Base Address + 0x00000040, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	
WORD_LENGTH	[3:1]	R	Word Length (field_size = 1) (field_size = 0) 000: not indicated not indicated 101: 24 bits 20 bits 100: 23 bits 19 bits 010: 22 bits 18 bits 110: 21 bits 17 bits	
FIELD_SIZE	[0]	R	Field Size 0: maximum length 20 bits 1: maximum length 24 bits	

SPDIFIN_FRAME_PERIOD_1

Address = Base Address + 0x00000048, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
FRAME_CNT_LO	[7:0]	R	frame count value [7:0] Frame_cnt register is 16bits value. This is low 8bits. The period of a frame (2 sub-frames), thus this register will be updated every 2 sub-frames. It will be measured by <i>SPDIFIN_internal_clk</i> made with SPDIFIN_CONFIG.clk_divisor. (Unit: SPDIF_internal_clk Cycles) (Recommended value for locking incoming signals : Over 0x220 (8.5timesx64bits))	

SPDIFIN_FRAME_PERIOD_2

Address = Base Address + 0x0000004C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
FRAME_CNT_HIGH	[7:0]	R	frame count value [15:8] Frame_cnt register is 16bits value. This is high 8bits. The period of a frame (2 sub-frames), thus this register will be updated every 2 sub-frames. It will be measured by <i>SPDIFIN_internal_clk</i> made with SPDIFIN_CONFIG.clk_divisor. (Unit: SPDIF_internal_clk Cycles) (Recommended value for locking incoming signals : Over 0x220 (8.5timesx64bits))	

SPDIFIN_Pc_INFO_1

Address = Base Address + 0x00000050, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ERROR_FLAG	[7]	R	0: valid burst payload 1: burst payload may contain errors	
RSVD	[6:5]	R	Reserved	

COMPRESSED_DATA_TYPE	[4:0]	R	0d: null data 1d: Dolby AC-3 2d: reserved 3d: pause 4d: MPEG-1 layer 1 5d: MPEG-1 layer 2 or 3 or MPEG-2 w/o extension 6d: MPEG-2 w/ extension 7d: reserved 8d: MPEG-2 layer 1 low sampling freq. 9d: MPEG-2 layer 2 or 3 low sampling freq. 10d: reserved	
-----------------------------	-------	---	--	--

SPDIFIN_Pc_INFO_2

Address = Base Address + 0x00000054, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BIT_STREAM_NUMBER	[7:5]	R	bit stream number.	
DATA_TYPE_DEPENDENT_INFO	[4:0]	R	data type dependent information.	

SPDIFIN_Pd_INFO_1

Address = Base Address + 0x00000058, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BURST_PAYLOAD_LENGTH_LO	[7:0]	R	length of burst payload [7:0] (Unit: bits)	

SPDIFIN_Pd_INFO_2

Address = Base Address + 0x0000005C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
BURST_PAYLOAD_LENGTH_HIG	[7:0]	R	length of burst payload [15:8] (Unit: bits)	

SPDIFIN_DATA_BUF_0_1

Address = Base Address + 0x00000060, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RECEIVED_DATA_0_1	[7:0]	R	PCM or stream data for 1st burst of HDMI SPDIFIN_DATA_BUF_0_1 = SPDIFIN_DATA_BUF_0[7:0] SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th} When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]} received_data = {zero-padding, data[n:0]} (if SPDIFIN_CONFIG.U_V_P_report is 0) (N is dependent on SPDIFIN_CH_STATUS_1. word length when SPDIFIN_CONFIG.data_type is 0 for PCM. Or N is 15 when SPDIFIN_CONFIG.data_type is 1 for stream) If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.	

SPDIFIN_DATA_BUF_0_2

Address = Base Address + 0x00000064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RECEIVED_DATA_0_2	[7:0]	R	<p>PCM or stream data for 1st burst of HDMI SPDIFIN_DATA_BUF_0_1 = SPDIFIN_DATA_BUF_0[7:0] SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8]</p> <p>SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th}</p> <p>When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]}</p> <p>received_data = {zero-padding, data[n:0]} (if SPDIFIN_CONFIG.U_V_P_report is 0)</p> <p>(N is dependent on SPDIFIN_CH_STATUS_1.word_length when SPDIFIN_CONFIG.data_type is 0 for PCM. Or N is 15 when SPDIFIN_CONFIG.data_type is 1 for stream)</p> <p>If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.</p>	

SPDIFIN_DATA_BUF_0_3

Address = Base Address + 0x00000068, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RECEIVED_DATA_0_3	[7:0]	R	<p>PCM or stream data for 1st burst of HDMI SPDIFIN_DATA_BUF_0_1 = SPDIFIN_DATA_BUF_0[7:0] SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8]</p> <p>SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th}</p> <p>When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]} received_data = {zero-padding, data[n:0]}</p> <p>(if SPDIFIN_CONFIG.U_V_P_report is 0)</p> <p>(N is dependent on SPDIFIN_CH_STATUS_1.word_length when SPDIFIN_CONFIG.data_type is 0 for PCM. Or N is 15 when SPDIFIN_CONFIG.data_type is 1 for stream)</p> <p>If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.</p>	

SPDIFIN_USER_BUF_0

Address = Base Address + 0x0000006C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RECEIVED_DATA_USER_0	[7:4]	R	User bit of 1st burst of HDMI received_data[7:4] = SPDIFIN_DATA_BUF_0[31:28]	
RSVD	[3:0]	R	Reserved	

SPDIFIN_DATA_BUF_1_1

Address = Base Address + 0x00000070, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

RECEIVED_DATA_DATA_1_1	[7:0]	R	PCM or stream data for 2nd burst of HDMI SPDIFIN_DATA_BUF_0[7:0] = SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] = SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th} When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]} received_data = {zero-padding, data[n:0]} (if SPDIFIN_CONFIG.U_V_P_report is 0) (N is dependent on SPDIFIN_CH_STATUS_1.word_length when SPDIFIN_CONFIG.data_type is 0 for PCM. Or N is 15 when SPDIFIN_CONFIG.data_type is 1 for stream) If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.
------------------------	-------	---	--

SPDIFIN_DATA_BUF_1_2

Address = Base Address + 0x000000074, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RECEIVED_DATA_DATA_1_2	[7:0]	R	PCM or stream data for 2nd burst of HDMI SPDIFIN_DATA_BUF_0[7:0] = SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] = SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th} When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]} received_data = {zero-padding, data[n:0]} (if SPDIFIN_CONFIG.U_V_P_report is 0) (N is dependent on SPDIFIN_CH_STATUS_1.word_length when SPDIFIN_CONFIG.data_type is 0 for PCM. Or N is 15 when SPDIFIN_CONFIG.data_type is 1 for stream) If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.	

SPDIFIN_DATA_BUF_1_3

Address = Base Address + 0x000000078, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RECEIVED_DATA_DATA_1_3	[7:0]	R	PCM or stream data for 2nd burst of HDMI SPDIFIN_DATA_BUF_0[7:0] = SPDIFIN_DATA_BUF_0_2 = SPDIFIN_DATA_BUF_0[15:8] = SPDIFIN_DATA_BUF_0_3 = SPDIFIN_DATA_BUF_0[23:16] When SPDIFIN_CONFIG.data_align is 0 for 16bit received_data = {data_(N)th, data_(N+1)th} When SPDIFIN_CONFIG.data_align is 1 for 32bit received_data = {U, V, C, P, zero-padding, data[n:0]} received_data = {zero-padding, data[n:0]} (if SPDIFIN_CONFIG.U_V_P_report is 0) (N is dependent on SPDIFIN_CH_STATUS_1.word_length when SPDIFIN_CONFIG.data_type is 0 for PCM. Or N is 15 when SPDIFIN_CONFIG.data_type is 1 for stream) If SPDIFIN_CONFIG.HDMI_burst_size is 1 burst, HDMI accesses only this data register.	

SPDIFIN_USER_BUF_1

Address = Base Address + 0x00000007C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RECEIVED_DATA_USER_1	[7:4]	R	User bit of 2nd burst of HDMI received_data[7:4] = SPDIFIN_DATA_BUF_1[31:28]	
RSVD	[3:0]	R	Reserved	

37.3.5 I2S Registers

Register Map Summary (Base Address 0xC024_0000)

Register	Offset	Description	Reset Value
I2S_CLK_CON	0x00000000	I2S Clock Enable Register	0x00
I2S_CON_1	0x00000004	I2S Control Register 1	0x00
I2S_CON_2	0x00000008	I2S Control Register 2	0x16
I2S_PIN_SEL_0	0x0000000C	I2S Input Pin Selection Register 0	0x77
I2S_PIN_SEL_1	0x00000010	I2S Input Pin Selection Register 1	0x77
I2S_PIN_SEL_2	0x00000014	I2S Input Pin Selection Register 2	0x77
I2S_PIN_SEL_3	0x00000018	I2S Input Pin Selection Register 3	0x07
I2S_DSD_CON	0x0000001C	I2S DSD Control Register	0x02
I2S_MUX_CON	0x00000020	I2S In/Mux Control Register	0x60
I2S_CH_ST_CON	0x00000024	I2S Channel Status Control Register	0x00
I2S_CH_ST_0	0x00000028	I2S Channel Status Block 0	0x00
I2S_CH_ST_1	0x0000002C	I2S Channel Status Block 1	0x00
I2S_CH_ST_2	0x00000030	I2S Channel Status Block 2	0x00
I2S_CH_ST_3	0x00000034	I2S Channel Status Block 3	0x00
I2S_CH_ST_4	0x00000038	I2S Channel Status Block 4	0x00
I2S_CH_ST_SH_0	0x0000003C	I2S Channel Status Block Shadow Register 0	0x00
I2S_CH_ST_SH_1	0x00000040	I2S Channel Status Block Shadow Register 1	0x00
I2S_CH_ST_SH_2	0x00000044	I2S Channel Status Block Shadow Register 2	0x00
I2S_CH_ST_SH_3	0x00000048	I2S Channel Status Block Shadow Register 3	0x00
I2S_CH_ST_SH_4	0x0000004C	I2S Channel Status Block Shadow Register 4	0x00
I2S_VD_DATA	0x00000050	I2S Audio Sample Validity Register	0x00
I2S_MUX_CH	0x00000054	I2S Channel Enable Register	0x03
I2S_MUX_CUV	0x00000058	I2S CUV Enable Register	0x03
I2S_CH0_L_0	0x00000064	I2S PCM Output Data Register	0x00
I2S_CH0_L_1	0x00000068	I2S PCM Output Data Register	0x00
I2S_CH0_L_2	0x0000006C	I2S PCM Output Data Register	0x00
I2S_CH0_R_0	0x00000074	I2S PCM Output Data Register	0x00
I2S_CH0_R_1	0x00000078	I2S PCM Output Data Register	0x00
I2S_CH0_R_2	0x0000007C	I2S PCM Output Data Register	0x00
I2S_CH0_R_3	0x00000080	I2S PCM Output Data Register	0x00
I2S_CH1_L_0	0x00000084	I2S PCM Output Data Register	0x00
I2S_CH1_L_1	0x00000088	I2S PCM Output Data Register	0x00
I2S_CH1_L_2	0x0000008C	I2S PCM Output Data Register	0x00

I2S_CH1_L_3	0x00000090	I2S PCM Output Data Register	0x00
I2S_CH1_R_0	0x00000094	I2S PCM Output Data Register	0x00
I2S_CH1_R_1	0x00000098	I2S PCM Output Data Register	0x00
I2S_CH1_R_2	0x0000009C	I2S PCM Output Data Register	0x00
I2S_CH1_R_3	0x000000A0	I2S PCM Output Data Register	0x00
I2S_CH2_L_0	0x000000A4	I2S PCM Output Data Register	0x00
I2S_CH2_L_1	0x000000A8	I2S PCM Output Data Register	0x00
I2S_CH2_L_2	0x000000AC	I2S PCM Output Data Register	0x00
I2S_CH2_L_3	0x000000B0	I2S PCM Output Data Register	0x00
I2S_CH2_R_0	0x000000B4	I2S PCM Output Data Register	0x00
I2S_CH2_R_1	0x000000B8	I2S PCM Output Data Register	0x00
I2S_CH2_R_2	0x000000BC	I2S PCM Output Data Register	0x00
I2S_CH2_R_3	0x000000C0	I2S PCM Output Data Register	0x00
I2S_CH3_L_0	0x000000C4	I2S PCM Output Data Register	0x00
I2S_CH3_L_1	0x000000C8	I2S PCM Output Data Register	0x00
I2S_CH3_L_2	0x000000CC	I2S PCM Output Data Register	0x00
I2S_CH3_R_0	0x000000D0	I2S PCM Output Data Register	0x00
I2S_CH3_R_1	0x000000D4	I2S PCM Output Data Register	0x00
I2S_CH3_R_2	0x000000D8	I2S PCM Output Data Register	0x00
I2S_CUV_L_R	0x000000DC	I2S CUV Output Data Register	0x00

I2S_CLK_CON

Address = Base Address + 0x00000000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
I2S_EN	[0]	RW	I2S Clock Enable 0: I2S will be disabled (default) 1: I2S will be activated You must set I2S_en, after other registers are configured. when you want to reset the I2S, this register is 0 - 1.	

I2S_CON_1

Address = Base Address + 0x00000004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
R_SC_POL	[1]	RW	SDATA is synchronous to 0: SCLK falling edge 1: SCLK rising edge	

R_CH_POL	[0]	RW	LRCLK polarity 0 : Left Channel for Low polarity 1 : Left Channel for High polarity	
-----------------	-----	----	---	--

I2S_CON_2

Address = Base Address + 0x00000008, Reset Value = 0x16

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	
MLSB	[6]	RW	0: MSB first mode 1: LSB first mode	
BIT_CH	[5:4]	RW	Bit clock per Frame(Frame = left + right) 0b00: 32fs 0b01: 48fs 0b10: 64fs	
DATA_NUM	[3:2]	RW	Serial data bit per channel 0b01: 16bit 0b10: 20bit 0b11: 24bit	
I2S_MODE	[1:0]	RW	0b00: I2S basic format 0b10: left justified format 0b11: right justified format	

I2S_PIN_SEL_0

Address = Base Address + 0x0000000C, Reset Value = 0x77

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	
PIN_SEL_1	[6:4]	RW	SCLK(I2S) & DSD_D0(DSD) selection 0b111 : i_I2S_in[1] 0b110 : i_I2S_in[6] 0b101 : i_I2S_in[5] 0b100 : i_I2S_in[4] 0b011 : i_I2S_in[3] 0b010 : i_I2S_in[2] 0b001 : i_I2S_in[1] 0b000 : i_I2S_in[0]	
RSVD	[3]	RW	Reserved	
PIN_SEL_0	[2:0]	RW	LRCK(I2S) & DSD_CLK(DSD) selection 0b111 : i_I2S_in[0] 0b110 : i_I2S_in[6] 0b101 : i_I2S_in[5] 0b100 : i_I2S_in[4] 0b011 : i_I2S_in[3] 0b010 : i_I2S_in[2] 0b001 : i_I2S_in[1] 0b000 : i_I2S_in[0]	

I2S_PIN_SEL_1

Address = Base Address + 0x00000010, Reset Value = 0x77

Name	Bit	Type	Description	Reset Value
------	-----	------	-------------	-------------

RSVD	[7]	RW	Reserved	
PIN_SEL_3	[6:4]	RW	SDATA_1(I2S) & DSD_D2(DSD) selection 0b111 : i_I2S_in[3] 0b110 : i_I2S_in[6] 0b101 : i_I2S_in[5] 0b100 : i_I2S_in[4] 0b011 : i_I2S_in[3] 0b010 : i_I2S_in[2] 0b001 : i_I2S_in[1] 0b000 : i_I2S_in[0]	
RSVD	[3]	RW	0	
PIN_SEL_2	[2:0]	RW	SDATA_0(I2S) & DSD_D1(DSD) selection 0b111 : i_I2S_in[2] 0b110 : i_I2S_in[6] 0b101 : i_I2S_in[5] 0b100 : i_I2S_in[4] 0b011 : i_I2S_in[3] 0b010 : i_I2S_in[2] 0b001 : i_I2S_in[1] 0b000 : i_I2S_in[0]	

I2S_PIN_SEL_2

Address = Base Address + 0x00000014, Reset Value = 0x77

Name	Bit	Type	Description	Reset Value
RSVD	[7]	RW	Reserved	
PIN_SEL_5	[6:4]	RW	SDATA_3(I2S) & DSD_D4(DSD) selection 0b111 : i_I2S_in[5] 0b110 : i_I2S_in[6] 0b101 : i_I2S_in[5] 0b100 : i_I2S_in[4] 0b011 : i_I2S_in[3] 0b010 : i_I2S_in[2] 0b001 : i_I2S_in[1] 0b000 : i_I2S_in[0]	
RSVD	[3]	RW	0	
PIN_SEL_4	[2:0]	RW	SDATA_2(I2S) & DSD_D3(DSD) selection 0b111 : i_I2S_in[4] 0b110 : i_I2S_in[6] 0b101 : i_I2S_in[5] 0b100 : i_I2S_in[4] 0b011 : i_I2S_in[3] 0b010 : i_I2S_in[2] 0b001 : i_I2S_in[1] 0b000 : i_I2S_in[0]	

I2S_PIN_SEL_3

Address = Base Address + 0x00000018, Reset Value = 0x07

Name	Bit	Type	Description	Reset Value
RSVD	[7:3]	RW	Reserved	
PIN_SEL_6	[2:0]	RW	DSD_D5(DSD) selection 0b111 : i_I2S_in[6] 0b110 : i_I2S_in[6] 0b101 : i_I2S_in[5] 0b100 : i_I2S_in[4] 0b011 : i_I2S_in[3] 0b010 : i_I2S_in[2] 0b001 : i_I2S_in[1] 0b000 : i_I2S_in[0]	

I2S_DSD_CON

Address = Base Address + 0x0000001C, Reset Value = 0x02

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
R_DSD_POL	[1]	RW	1 : DSD_DATA change at DSD_CLK rising edge 0 : DSD_DATA change at DSD_CLK falling edge	
DSD_EN	[0]	RW	1 : DSD module enable 0 : DSD module disable	

I2S_MUX_CON

Address = Base Address + 0x00000020, Reset Value = 0x60

Name	Bit	Type	Description	Reset Value
F_NUM	[7:5]	RW	Number of stage of noise filter for I2S input pins 000: no filtering 001: 2 stage filter 010: 3 stage filter 011: 4 stage filter	
IN_EN	[4]	RW	Enable I2S_in, a sub-module at the input stage. 0 : I2S_in module disable 1 : I2S_in module enable All output data is 0 if disabled.	
AUDIO_SEL	[3:2]	RW	Audio selection 0b00 : SPDIF audio data enable 0b01 : I2S audio data enable 0b10 : DSD audio data enable	
CUV_SEL	[1]	RW	C.U.V. Selection 0 : SPDIF C.U.V. data enable 1 : I2S C.U.V. data enable	
MUX_EN	[0]	RW	Enable I2S_mux, a sub-module for audio selection. 0 : I2S_mux module disable 1 : I2S_mux module enable All output data is 0 if disabled.	

I2S_CH_ST_CON

Channel status information needs to be applied to the audio stream at the IEC 60958 block boundary. For this synchronization, there are two register sets for channel status block. Users can set the channel status registers, I2S_CH_ST_0~I2S_CH_ST_4, while the I2S Rx module still refers to the shadow channel status registers, I2S_CH_ST_SH_0~I2S_CH_ST_CH4. To reflect the user configuration in the channel status registers, users should set *channel_status_reload* bit in I2S_CH_ST_CON then I2S Rx module copies the channel status registers into the shadow channel status registers at the beginning of an IEC-60958 block.

Address = Base Address + 0x00000024, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
CHANNEL_STATUS_RELOAD	[0]	RW	0: The shadow channel status registers are updated. 1: Set this bit to update the shadow channel status registers with the values updated in I2S_CH_ST_0 ~ I2S_CH_ST_4. When the shadow channel status registers are updated, this bit is cleared.	

I2S_CH_ST_0

Note that bits listed here in Channel Status Registers look swapped from those in IEC-60958-3 Specification, as the bit order is different (LSB is right-most bit)

Address = Base Address + 0x00000028, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
HANNEL_STATUS_MODE	[7:6]	RW	0b00 : Mode 0 others : Reserved	
EMPHASIS	[5:3]	RW	When bit1 = 0, 0b000 : 2 audio channels without pre-emphasis* 0b001 : 2 audio channels with 50us / 15us pre-emphasis When bit1 = 1, 0b000 : default state	
COPYRIGHT	[2]	RW	0: copyright 1: no copyright	
AUDIO_SAMPLE_WORD	[1]	RW	0: linear PCM 1: non-linear PCM	
CHANNEL_STATUS_BLOC K	[0]	RW	0: consumer format 1: professional format	

I2S_CH_ST_1

Address = Base Address + 0x0000002C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CATEGORY	[7:0]	RW	Equipment type CD player : 0000_0001 DAT player : L000_0011 DCC player : L100_0011 Mini disc : L100_1001	

I2S_CH_ST_2

Address = Base Address + 0x00000030, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CHANNEL_NUMBER	[7:4]	RW	Channel Number Note that bit4 is LSB.	
SOURCE_NUMBER	[3:0]	RW	Source Number Note that bit0 is LSB.	

I2S_CH_ST_3

Address = Base Address + 0x00000034, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	RW	Reserved	

CLOCK_ACCURACY	[5:4]	RW	Clock Accuracy as specified in IEC-60958-3 0b01 : Level I, ±50 ppm 0b00 : Level II, ±1000 ppm 0b10 : Level III, variable pitch shifted	
SAMPLING_FREQUENCY	[3:0]	RW	Sampling Frequency as specified in IEC-60958-3 0b0000 : 44.1 kHz 0b0010 : 48 kHz 0b0011 : 32 kHz 0b1010 : 96 kHz &	

I2S_CH_ST_4

Address = Base Address + 0x00000038, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
ORG_SAMPLING_FREQ	[7:4]	RW	Original Sampling Frequency 0b1111 : 44.1Khz 0b0111 : 88.2Khz 0b1011 : 22.05Khz 0b0011 : 176.4Khz & For other frequencies, refer to original sampling frequency specified in	
WORD_LENGTH	[3:1]	RW	Word length Max. length 24bits 20 bits 0b000 : not defined not defined 0b001 : 20 bits 16bits 0b010 : 22 bits 18bits 0b100 : 23 bits 19bits 0b101 : 24 bits 20bits	
MAX_WORD_LENGTH	[0]	RW	Maximum sample word length 1 : 24 bits 0 : 20 bits	

I2S_CH_ST_SH_0

Address = Base Address + 0x0000003C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CHANNEL_STATUS_MODE	[7:6]	R	0b00 : Mode 0 others : Reserved	
EMPHASIS	[5:3]	R	When bit1 = 0, 0b000 : 2 audio channels without pre-emphasis* 0b001 : 2 audio channels with 50us / 15us pre-emphasis When bit1 = 1, 0b000 : default state	
COPYRIGHT	[2]	R	0 : copyright 1 : no copyright	
AUDIO_SAMPLE_WORD	[1]	R	0 : linear PCM 1 : non-linear PCM	

CHANNEL_STATUS_BLOC_K	[0]	R	0 : consumer format 1 : professional format	
-----------------------	-----	---	--	--

I2S_CH_ST_SH_1

Address = Base Address + 0x00000040, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CATEGORY	[7:0]	R	Equipment type CD player : 0000_0001 DAT player : L000_0011 DCC player : L100_0011 Mini disc : L100_1001 (L : information about generation status of the material)	

I2S_CH_ST_SH_2

Address = Base Address + 0x00000044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CHANNEL_NUMBER	[7:4]	R	Channel Number Note that bit4 is LSB.	
SOURCE_NUMBER	[3:0]	R	Source Number Note that bit0 is LSB.	

I2S_CH_ST_SH_3

Address = Base Address + 0x00000048, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	R	Reserved	
CLOCK_ACCURACY	[5:4]	R	Clock Accuracy as specified in IEC-60958-3 0b01 : Level I, ±50 ppm 0b00 : Level II, ±1000 ppm 0b10 : Level III, variable pitch shifted	
SAMPLING_FREQUENCY	[3:0]	R	Sampling Frequency as specified in IEC-60958-3 0b0000 : 44.1 kHz 0b0010 : 48 kHz 0b0011 : 32 kHz 0b1010 : 96 kHz &	

I2S_CH_ST_SH_4

Address = Base Address + 0x0000004C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

ORG_SAMPLING_FREQ	[7:4]	RW	Original Sampling Frequency 0b1111 : 44.1Khz 0b0111 : 88.2Khz 0b1011 : 22.05Khz 0b0011 : 176.4Khz & For other frequencies, refer to original sampling frequency specified in IEC-60958-3	
WORD_LENGTH	[3:1]	RW	Word length Max. length 24bits 20 bits 0b000 : not defined not defined 0b001 : 20 bits 16bits 0b010 : 22 bits 18bits 0b100 : 23 bits 19bits 0b101 : 24 bits 20bits	
MAX_WORD_LENGTH	[0]	RW	Maximum sample word length 1 : 24 bits 0 : 20 bits	

I2S_VD_DATA

Address = Base Address + 0x00000050, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:1]	RW	Reserved	
VALIDITY_FLAG	[0]	RW	Validity bit 0 : audio sample is reliable 1 : audio sample is unreliable	

I2S_MUX_CH

Address = Base Address + 0x00000054, Reset Value = 0x03

Name	Bit	Type	Description	Reset Value
CH3_R_EN	[7]	RW	0 : Channel 3 right audio data output is disable 1 : Channel 3 right audio data output is enable	
CH3_L_EN	[6]	RW	0 : Channel 3 left audio data output is disable 1 : Channel 3 left audio data output is enable	
CH2_R_EN	[5]	RW	0 : Channel 2 right audio data output is disable 1 : Channel 2 right audio data output is enable	
CH2_L_EN	[4]	RW	0 : Channel 2 left audio data output is disable 1 : Channel 2 left audio data output is enable	
CH1_R_EN	[3]	RW	0 : Channel 1 right audio data output is disable 1 : Channel 1 right audio data output is enable	
CH1_L_EN	[2]	RW	0 : Channel 1 left audio data output is disable 1 : Channel 1 left audio data output is enable	
CH0_R_EN	[1]	RW	0 : Channel 0 right audio data output is disable 1 : Channel 0 right audio data output is enable	

CH0_L_EN	[0]	RW	0: Channel 0 left audio data output is disable 1: Channel 0 left audio data output is enable	
-----------------	-----	----	---	--

I2S_MUX_CUV

Address = Base Address + 0x00000058, Reset Value = 0x03

Name	Bit	Type	Description	Reset Value
RSVD	[7:2]	RW	Reserved	
CUV_R_EN	[1]	RW	0: Right channel CUV data is disable 1: Right channel CUV data is enable	
CUV_L_EN	[0]	RW	0: Left channel CUV data is disable 1: Left channel CUV data is enable	

I2S_CH0_L_0

Address = Base Address + 0x00000064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH0_L_1

Address = Base Address + 0x00000068, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH0_L_2

Address = Base Address + 0x0000006C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH0_R_0

Address = Base Address + 0x00000074, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH0_R_1

Address = Base Address + 0x00000078, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH0_R_2

Address = Base Address + 0x0000007C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH0_R_3

Address = Base Address + 0x00000080, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH1_L_0

Address = Base Address + 0x00000084, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH1_L_1

Address = Base Address + 0x00000088, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH1_L_2

Address = Base Address + 0x0000008C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH1_L_3

Address = Base Address + 0x00000090, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH1_R_0

Address = Base Address + 0x00000094, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH1_R_1

Address = Base Address + 0x00000098, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	
-------------	-------	---	--	--

I2S_CH1_R_2

Address = Base Address + 0x00000009C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH1_R_3

Address = Base Address + 0x0000000A0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH2_L_0

Address = Base Address + 0x0000000A4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH2_L_1

Address = Base Address + 0x0000000A8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH2_L_2

Address = Base Address + 0x000000AC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH2_L_3

Address = Base Address + 0x000000B0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH2_R_0

Address = Base Address + 0x000000B4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH2_R_1

Address = Base Address + 0x000000B8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH2_R_2

Address = Base Address + 0x000000BC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	
-------------	-------	---	--	--

I2S_Ch2_R_3

Address = Base Address + 0x000000C0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH3_L_0

Address = Base Address + 0x000000C4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH3_L_1

Address = Base Address + 0x000000C8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH3_L_2

Address = Base Address + 0x000000CC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X_Y[7:0] I2S_CHX_Y_1 = PCM_X_Y[15:8] I2S_CHX_Y_2 = PCM_X_Y[23:16] I2S_CHX_Y_3 = PCM_X_Y[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH3_R_0

Address = Base Address + 0x000000D0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH3_R_1

Address = Base Address + 0x000000D4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CH3_R_2

Address = Base Address + 0x000000D8, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
I2S_CHX_Y_Z	[7:0]	R	PCM output data from I2S Rx module. X : Channel = 0, 1, 2 Y : Left/Right = L, R Z : Byte number I2S_CHX_Y_0 = PCM_X[7:0] I2S_CHX_Y_1 = PCM_X[15:8] I2S_CHX_Y_2 = PCM_X[23:16] I2S_CHX_Y_3 = PCM_X[27:24] Channel 3 has 24 bitwidth. I2S_CH3_Y_0 = PCM3_Y[7:0] I2S_CH3_Y_1 = PCM3_Y[15:8] I2S_CH3_Y_2 = PCM3_Y[23:16]	

I2S_CUV_L_R

Reserved

Address = Base Address + 0x000000DC, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	
CUV_R	[6:4]	RW	VUCP data of Right channel CUV_R[3:0] = {valid bit, user bit, channel state bit, parity bit}	
RSVD	[3]	RW	Reserved	
CUV_L	[2:0]	RW	VUCP data of Left channel CUV_L[3:0] = {valid bit, user bit, channel state bit, parity bit}	

37.3.6 CEC Registers

Register Map Summary (Base Address 0xC010_0000)

Register	Offset	Description	Reset Value

CEC_TX_STATUS_0	0x00000000	CEC Tx status register 0.	0x00
CEC_TX_STATUS_1	0x00000004	CEC Tx status register 1. Number of blocks transferred.	0x00
CEC_RX_STATUS_0	0x00000008	CEC Rx status register 0.	0x00
CEC_RX_STATUS_1	0x0000000C	CEC Rx status register 1. Number of blocks received.	0x00
CEC_INTR_MASK	0x00000010	CEC interrupt mask register	0x00
CEC_INTR_CLEAR	0x00000014	CEC interrupt clear register	0x00
CEC_LOGIC_ADDR	0x00000020	HDMI Tx logical address register	0x0F
CEC_DIVISOR_0	0x00000030	Clock divisor for 0.05ms period count ([7:0] of 32-bit)	0x00
CEC_DIVISOR_1	0x00000034	Clock divisor for 0.05ms period count ([15:8] of 32-bit)	0x00
CEC_DIVISOR_2	0x00000038	Clock divisor for 0.05ms period count ([23:16] of 32-bit)	0x00
CEC_DIVISOR_3	0x0000003C	Clock divisor for 0.05ms period count ([31:24] of 32-bit)	0x00
CEC_TX_CTRL	0x00000040	CEC Tx control register	0x10
CEC_TX_BYTE_NUM	0x00000044	Number of blocks in a message to be transferred	0x00
CEC_TX_STATUS_2	0x00000060	CEC Tx status register 2	0x00
CEC_TX_STATUS_3	0x00000064	CEC Tx status register 3	0x00
CEC_RX_BUFFER_X	0x00000080	Byte #0 ~ #15 of CEC message to be transferred. (#0 is transferred 1st)	0x00
CEC_RX_CTRL	0x000000C0	CEC Rx control register	0x00
CEC_RX_STATUS_2	0x000000E0	CEC Rx status register 2	0x00
CEC_RX_STATUS_3	0x000000E4	CEC Rx status register 3	0x00
CEC_RX_BUFFER_X	0x00000100	Byte #0 ~ #15 of CEC message received (#0 is received 1st)	0x00
CEC_FILTER_CTRL	0x00000180	CEC Filter control register	0x81
CEC_FILTER_TH	0x00000184	CEC Filter Threshold register	0x03

CEC_TX_STATUS_0

Address = Base Address + 0x00000000, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	
TX_ERROR	[3]	R	CEC Tx_Error interrupt flag. This bit field also indicates the status of Tx_Error interrupt. This bit is valid only when Tx_Done bit is set. O : No error has occurred. 1 : An error has occurred during CEC Tx transfer. It will be cleared: when set to 0 Tx_Enable bit of CEC_TX_CTRL register when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_INTR_CLEAR register	
TX_DONE	[2]	R	CEC Tx_Done interrupt flag. This bit field also indicates the status of Tx_Done interrupt. O : Running or Idle 1 : CEC Tx transfer finished. It will be cleared: <ul style="list-style-type: none">• when reset Tx_Enable bit of CEC_TX_CTRL_0• when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_INTR_CLEAR register	
TX_TRANSFERRING	[1]	R	If set RX-Running , this field is valid O : Tx is waiting for CEC Bus 1 : CEC Tx is transferring data via CEC Bus.	

TX_RUNNING	[0]	R	O : Tx Idle 1 : CEC Tx is enabled and is either waiting for the CEC bus or transferring message.	
-------------------	-----	---	---	--

CEC_TX_STATUS_1

Address = Base Address + 0x00000004, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
TX_BYTES_TRANSFERRED	[7:0]	R	Number of blocks transferred (1 byte = 1 block in a CEC message). After sending CEC message, field will be updated. It will be cleared when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_Intr_Clear register.	

CEC_RX_STATUS_0

Address = Base Address + 0x00000008, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:5]	R	Reserved	
RX_BCAST	[4]	R	Broadcast message flag O : Received CEC message is address to a single device. 1 : Received CEC message is a broadcast message. It will be cleared: <ul style="list-style-type: none">• when reset Rx_Enable bit of CEC_RX_CTRL_0• when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register	
RX_ERROR	[3]	R	CEC Rx_Error interrupt flag. This bit field also indicates the status of Rx_Error interrupt. This bit is valid only when Rx_Done bit is set. O : No error has occurred. 1 : An error has occurred in receiving a CEC message It will be cleared: <ul style="list-style-type: none">• when reset Rx_Enable bit of CEC_RX_CTRL_0• when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register	
RX_DONE	[2]	R	CEC Rx done interrupt Flag. This bit field also indicates the status of Rx_Done interrupt. O : Running or Idle 1 : CEC Rx transfer finished It will be cleared: <ul style="list-style-type: none">• when reset Rx_Enable bit of CEC_RX_CTRL_0• when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register	
RX RECEIVING	[1]	R	O : Rx is waiting for a CEC message. 1 : Rx is currently receiving data via CEC Bus.	
RX_RUNNING	[0]	R	O : Rx disabled 1 : CEC Rx is enabled and is either waiting for a message on the CEC bus.	

CEC_RX_STATUS_1

Address = Base Address + 0x0000000C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
------	-----	------	-------------	-------------

RX_BYTES_RECEIVED	[7:0]	R	Number of blocks received (1 byte = 1 block in a CEC message). After receiving CEC message, field will be updated. It will be cleared when set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_Intr_Clear register.	
--------------------------	-------	---	---	--

CEC_INTR_MASK

Address = Base Address + 0x00000010, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	R	Reserved	
MASK_INTR_RX_ERROR	[5]	RW	Rx_Error interrupt mask bit. 0 : Enabled 1 : Disabled.	
MASK_INTR_RX_DONE	[4]	RW	Rx_Done interrupt mask bit. 0 : Enabled 1 : Disabled.	
RSVD	[3:2]	RW	Reserved	
MASK_INTR_TX_ERROR	[1]	RW	Tx_Error interrupt mask bit. 0 : Enabled 1 : Disabled.	
MASK_INTR_TX_DONE	[0]	RW	Tx_Done interrupt mask bit. 0 : Enabled 1 : Disabled.	

CEC_INTR_CLEAR

Address = Base Address + 0x00000014, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7:6]	R	Reserved	
CLEAR_INTR_RX_ERROR	[5]	R/W1C	Rx_Error interrupt clear bit. Writing following values will result in: 0 : No effect 1 : Clear Rx_Error and Rx_Bytes_Received fields in CEC_RX_STATUS_0 and 1 register. It will be cleared after one clock.	
CLEAR_INTR_RX_DONE	[4]	R/W1C	Rx_Done interrupt clear bit. Writing following values will result in: 0 : No effect 1 : Clears Rx_Done and Rx_Bytes_Received fields in CEC_RX_STATUS_0 and 1 register. Resets to 0 after one clock.	
RSVD	[3:2]	R	Reserved	
CLEAR_INTR_TX_ERROR	[1]	R/W1C	Tx_Error interrupt clear bit. Writing following values will result in: 0 : No effect 1 : Clears Tx_Error and Tx_Bytes_Received fields in CEC_TX_STATUS_0 and 1 register. Resets to 0 after one clock.	

CLEAR_INTR_TX_DONE	[0]	R/W1C	Tx_Done interrupt clear bit. Writing following values will result in: 0 : No effect 1 : Clears Tx_Done and Tx_Bytes_Received fields in CEC_TX_STATUS_0 and 1 register. Resets to 0 after one clock.	
--------------------	-----	-------	---	--

CEC_LOGIC_ADDR

Address = Base Address + 0x00000020, Reset Value = 0x0F

Name	Bit	Type	Description	Reset Value
RSVD	[7:4]	R	Reserved	
LOGIC_ADDR	[3:0]	RW	HDMI Tx logical address (0-15)	

CEC_DIVISOR_0

Address = Base Address + 0x00000030, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CEC_DIVISOR	[7:0]	RW	(CEC_Divisor[7:0] of 32-bit) A divisor used in counting 0.05ms period. This divisor should satisfy the following equation: (CEC_DIVISOR) x (clock cycle time(ns)) = 0.05ms	

CEC_DIVISOR_1

Address = Base Address + 0x00000034, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CEC_DIVISOR	[7:0]	RW	(CEC_Divisor[15:8] of 32-bit) A divisor used in counting 0.05ms period. This divisor should satisfy the following equation: (CEC_DIVISOR) x (clock cycle time(ns)) = 0.05ms	

CEC_DIVISOR_2

Address = Base Address + 0x00000038, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
CEC_DIVISOR	[7:0]	RW	(CEC_Divisor[23:16] of 32-bit) A divisor used in counting 0.05ms period. This divisor should satisfy the following equation: (CEC_DIVISOR) x (clock cycle time(ns)) = 0.05ms	

CEC_DIVISOR_3

Address = Base Address + 0x0000003C, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value

CEC_DIVISOR	[7:0]	RW	(CEC_Divisor[31:24] of 32-bit) A divisor used in counting 0.05ms period. This divisor should satisfy the following equation: (CEC_DIVISOR) x (clock cycle time(ns)) = 0.05ms	
--------------------	-------	----	--	--

CEC_TX_CTRL

When Reset field is set to 1, CEC_TX_CTRL, CEC_TX_STATUS_0~3, CEC_TX_BUFFER0~15 will be set to their reset values.

Address = Base Address + 0x00000040, Reset Value = 0x10

Name	Bit	Type	Description	Reset Value
RESET	[7]	R/W1C	CEC Tx reset bit. Writing following values will result in: 0 : No effect 1 : Immediately resets CEC Tx related registers and state machines to its reset value. Resets to 0 after one clock.	
TX_RETRANS_NUM	[6:4]	RW	Number of retransmissions tried when situations in CEC spec. page CEC-13 occurs. According to the specification, it should be set to 5.	
RSVD	[3:2]	R	Reserved	
TX_BCAST	[1]	RW	CEC Tx broadcast message bit. This bit indicates whether a CEC message in CEC_TX_BUFFER_00~15 is directly-addressed (addressed to a single device) or broadcast. This bit has effect on determining whether a block transfer is acknowledged or not. (following ACK scheme in CEC Spec.(section CEC 6.1.2)) 0 : Directly-addressed message 1 : Broadcast message.	
TX_START	[0]	R/W1C	CEC Tx start bit. Writing following values will result in: 0 : Tx idle. 1 : Start CEC message transfer (Resets to 0 after start)	

CEC_TX_BYTE_NUM

Address = Base Address + 0x00000044, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
TX_BYTE_NUM	[7:0]	RW	Number of blocks in a message to be sent. (1 byte = 1 block in a CEC message).	

CEC_TX_STATUS_2

Address = Base Address + 0x00000060, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
TX_WAIT	[7]	R	CEC Tx signal free time waiting flag bit 0 : Tx is in other state 1 : CEC Tx is waiting for a signal free time(time to stop sending messages after previous attempt to send a message).	
TX_SENDING_START_BIT	[6]	R	CEC Tx start bit sending flag bit 0 : Tx is in other state 1 : CEC Tx is currently sending a start bit.	

TX_SENDING_HDR_BLK	[5]	R	CEC Tx header block sending flag bit 0 : Tx is in other state 1 : CEC Tx is currently sending the header block.	
TX_SENDING_DATA_BLK	[4]	R	CEC Tx data block sending flag bit 0 : Tx is in other state 1 : CEC Tx is currently sending data blocks.	
TX_LATEST_INITIATOR	[3]	R	CEC Tx last initiator flag bit 0 : This device is not the latest initiator on the CEC bus. 1 : This CEC device is the latest initiator to send a CEC message and no other CEC device sent a message. It will be cleared if Rx detects a start bit on the CEC line or Tx_Enable bit of CEC_Tx_Ctrl_0 is set (i.e. becomes a new initiator)	
RSVD	[2:0]	R	Reserved	

CEC_TX_STATUS_3

Address = Base Address + 0x00000064, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	
TX_WAIT_SFT_SUCC	[6]	R	CEC Tx signal free time for successive message transfer waiting flag bit 0 : Tx is in other state 1 : Tx is waiting for a signal free time (SFT) with a precondition that Tx is the most recent initiator on the CEC bus and wants to send another frame immediately after its previous frame. (SFT ≥ 7x2.4ms)	
TX_WAIT_SFT_NEW	[5]	R	CEC Tx signal free time for a new initiator waiting flag bit 0 : Tx is in other state 1 : Tx is waiting for a signal free time (SFT) with a precondition that Tx is a new initiator and wants to send a frame. (SFT ≥ 5x2.4ms)	
TX_WAIT_SFT_RETRAN	[4]	R	CEC Tx signal free time for a new initiator waiting flag bit 0 : Tx is in other state 1 : Tx is waiting for a signal free time (SFT) with a precondition that Tx is attempting a retransmission of the message. (SFT ≥ 3 x2.4ms)	
TX_RETRANS_CNT	[3:1]	R	It indicates current retransmissions count. If 0, no retransmission occurred. It will be cleared when set Clear_Intr_Tx_Done or Clear_Intr_Tx_Error bit in CEC_Intr_Clear register.	
TX_ACK_FAILED	[0]	R	CEC Tx acknowledge failed flag bit 0 : Tx is in other state 1 : Tx is not acknowledged. This bit is set when <ul style="list-style-type: none"> • ACK bit in a block is logical 1 in a directly-addressed message • ACK bit in a block is logical 0 in a broadcast message 	

CEC_TX_BUFFER_X

Address = Base Address + 0x00000080, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
TX_BLOCK_X	[7:0]	RW	Byte #0 ~ #15 of CEC message. Each byte corresponds to a block in a message. Block_0 is the header block and Block_1 ~15 are data blocks. Note that initiator and destination logical address in a header block should be written by S/W.	

CEC_RX_CTRL

When Reset field is set to 1, CEC_RX_CTRL, CEC_RX_STATUS_0~3, CEC_RX_BUFFER0~15 will be set to their reset values.

Address = Base Address + 0x000000C0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RESET	[7]	RW1C	CEC Rx reset bit. Writing following values will result in: 0 : No effect 1 : Immediately resets CEC Rx related registers and state machines to its reset value. It will be cleared after one clock.	
CHECK_SAMPLING_ERROR	[6]	RW	CEC Rx sampling error check enable bit. Writing following values will result in: 0 : Do not check sampling error. 1 : Check sampling error while receiving data bits. CEC Rx samples the CEC bus three times (at 1.00, 1.05, 1.10 ms) and checks whether three samples are identical.	
CHECK_LOW_TIME_ERROR	[5]	RW	CEC Rx low-time error check enable bit. Writing following values will result in: 0 : Do not check low-time error. 1 : Check low-time error while receiving data bits. In receiving each bit from the CEC bus, CEC Rx checks the duration of logical 0 from the starting of one bit transfer (falling edge on the CEC bus). Rx checks whether the duration is longer than the maximum time the CEC bus can be in logical 0 (max 1.7 ms).	
CHECK_START_BIT_ERROR	[4]	RW	CEC Rx start bit error check enable bit. Writing following values will result in: 0 : Do not check start bit error. 1 : Check start bit error while receiving a start bit. In receiving a start bit from the CEC bus, CEC Rx checks the duration of logical 0 and 1 of a starting bit as specified in CEC spec page CEC-8. Rx checks whether the duration meets the specification.	
RSVD	[3:2]	R	Reserved	
RX_HOST_BUSY	[1]	RW	CEC Rx host busy bit. Writing following values will result in: 0 : Rx receives incoming message and send acknowledges. 1 : A host processor is unavailable to receive and process CEC messages. Rx sends not acknowledged signal to a message initiator to indicate that a host processor is unavailable to receive and process CEC messages.	
RX_ENABLE	[0]	RW	CEC Rx start bit. Writing following values will result in: 0 : Rx disabled. 1 : Enable CEC Rx module to receive a message. This bit is cleared after receiving a message.	

CEC_RX_STATUS_2

Address = Base Address + 0x000000E0, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RX_WAITING	[7]	R	CEC Rx waiting flag bit 0 : Rx is in other state 1 : CEC Rx is waiting for a message.	
RX RECEIVING START BIT	[6]	R	CEC Rx start bit receiving flag bit 0 : Rx is in other state 1 : CEC Rx is currently receiving a start bit.	
RX RECEIVING HDR_BLK	[5]	R	CEC Rx header block receiving flag bit 0 : Rx is in other state 1 : CEC Rx is currently receiving a header block.	
RX RECEIVING DATA_BLK	[4]	R	CEC Rx data block receiving flag bit 0 : Rx is in other state 1 : CEC Rx is currently receiving data blocks.	
RSVD	[3:0]	R	Reserved	

CEC_RX_STATUS_3

Address = Base Address + 0x000000E4, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RSVD	[7]	R	Reserved	
SAMPLING_ERROR	[6]	R	CEC Rx sampling error flag bit 0 : No sampling error has occurred. 1 : A sampling error has occurred in receiving a message. CEC Rx samples the CEC bus three times (at 1.00, 1.05, 1.10 ms) and sets this bit if <ul style="list-style-type: none"> • Check_Sampling_Error bit in CEC_RX_CTRL_0 is set and • Three samples are not identical. It will be cleared when set to 0 when Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register.	
LOW_TIME_ERROR	[5]	R	CEC Rx low-time error flag bit 0 : No low-time error has occurred. 1 : A low-time error has occurred in receiving a message. In receiving each bit from the CEC bus, CEC Rx checks the duration of logical 0 from the starting of one-bit transfer (falling edge on the CEC bus). If the duration is longer than the maximum time the CEC bus can be in logical 0 (max 1.7 ms), CEC RX sets this bit. This bit field will be set to 0 when Clear_Intr_Rx_Done or	
START_BIT_ERROR	[4]	R	CEC Rx start bit error flag bit 0 : No start bit error has occurred. 1 : A start bit error has occurred in receiving a message. In receiving a start bit from the CEC bus, CEC Rx checks the duration of logical 0 and 1 of a starting bit as specified in CEC spec. page CEC-8. If the duration does not meet the spec., CEC RX sets this bit. This bit field will be set to 0 when Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register is set.	
RSVD	[3:1]	R	Reserved	

CEC_LINE_ERROR	[0]	R	<p>CEC Rx line error error flag bit</p> <p>0 : No line error has occurred.</p> <p>1 : A start bit error line error has occurred in receiving a message.</p> <p>In CEC spec. page CEC-13, CEC line error is defined as a situation that period between two consecutive falling edge is smaller than a minimum data bit period. Rx check for this condition and if it occurs, sends line error notification, i.e. sending logical 0 for more than 1.4~1.6 times of the nominal data bit period (2.4ms).</p> <p>This bit will be cleared:</p> <ul style="list-style-type: none"> • When set Rx_Enable bit of CEC_RX_CTRL_0 • When set Clear_Intr_Rx_Done or Clear_Intr_Rx_Error bit in CEC_INTR_CLEAR register. 	
-----------------------	-----	---	--	--

CEC_RX_BUFFER_x

Address = Base Address + 0x00000100, Reset Value = 0x00

Name	Bit	Type	Description	Reset Value
RX_BLOCK_X	[7:0]	R	Byte #0 ~ #15 of CEC message. Each byte corresponds to a block in a message. Block_0 is the header block and Block_1 ~ 15 are data blocks.	

CEC_FILTER_CTRL

Address = Base Address + 0x00000180, Reset Value = 0x81

Name	Bit	Type	Description	Reset Value
FILTER_CUR_VAL	[7]	RW	CEC filter current value bit. Indicates current value fed to CEC Tx, Rx. When filter is enabled, this bit is the latest value on the CEC bus that is stable for more than Filter_Th cycles.	
RSVD	[6:1]	R	Reserved	
FILTER_ENABLE	[0]	RW	<p>CEC filter enable bit.</p> <p>0 : Filter disabled. Directly passes CEC input to CEC Tx, Rx.</p> <p>1 : Enable Filter. Filter propagates signals stable for more Filter_Th cycles.</p>	

CEC_FILTER_TH

Address = Base Address + 0x00000184, Reset Value = 0x03

Name	Bit	Type	Description	Reset Value
FILTER_TH	[7:0]	RW	Filter threshold value. When filter is enabled, it filters out signals stable for less than Filter_Th cycles	

37.4 HDMI PHY

The HDMI PHY can generate a pixel clock for HDMI 1.4 spec with own PCG (Pixel clock generator) that used 24 MHz reference clock. Following is generatable pixel clock frequency table.

Available pixel clock frequency for DTV (Mhz)	Available pixel clock frequency for Monitor (Mhz)
25.2	25
25.175	65
27	108
27.027	162
54	
54.054	
74.25	
74.176	
148.5	
148.352	
108.108	
72	

Table 37-2. Available Pixel Clock Frequencies of the Integrated Video PLL

37.4.1 PHY Configuration Change Through APB

The HDMI PHY has many internal registers to change its configuration, like pixel clock frequency or analog characteristics. Users can access these registers through APBport. For secure configuration of the PHY core, MODE_SET_DONE register(REG_7C<7>)is used for an indicator of APB setting state as shown in Figure 37-16. (MODE_SET_DONE register is also controlled by APB.)

If users wants to reconfigure the HDMI PHY by new register setting, it should write 00h on **MODE_SET_DONE** register(0xC010047C) instead of asserting overall RESET signal. Then PHY_READY signal goes to low state and PHY waits for new register setting. After new values are written on PHY registers, MODE_SET_DONE register should be set to 80h again letting PHY start to configure its state with new register values. Once configuration is done, PHY_READY signal is automatically asserted.

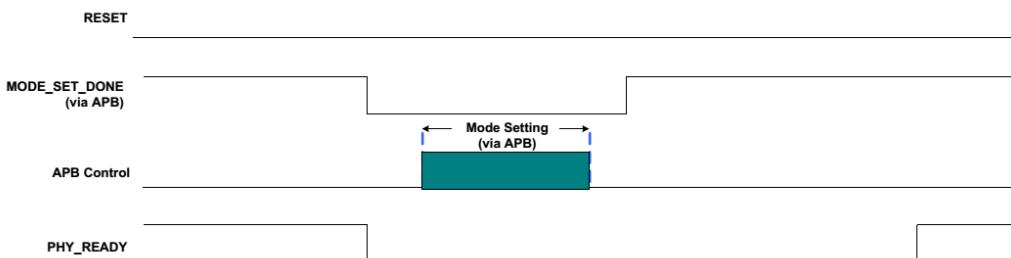


Figure 37-16. PHY configuration through APBwith MODE_SET_DONE register

37.4.2 PHY READY SEQUENCE

To assert PHY_READY signal, the HDMI PHY have to proceed several steps shown in Figure 37-17. The HDMI PHY core has a CMU and PCG for TMDS and pixel clock generation. Both of them should be locked and clock deskewing between TMDS_CLKHI and TMDS_CLKO should be finished before PHY_READY assertion. Thus, TMDS_CLKHI should be supplied to PHY before PHY_READY signal assertion. The HDMI PHY ready sequence is ignited by PHY_RESET signal which is external reset or inversion of MODE_SET_DONE signal. Once PLL_LOCK and CMU_LOCK signals go high, the deskewing process starts. After deskewing is finished, the PHY_READY signal goes high meaning The HDMI PHY is ready to correctly send TMDS data.

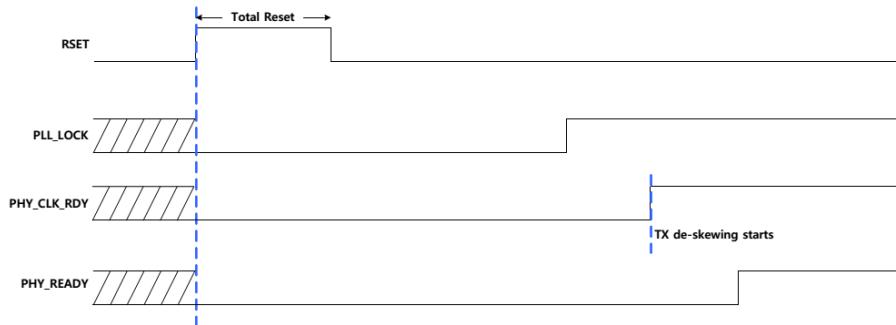


Figure 37-17. PHY READY sequence

37.4.3 HDMI PHY configuration

Users need to set the HDMI PHY configuration to get a generated pixel clock. Following is a sequence of setting the HDMI PHY configuration.

- 1) Set the HDMI CLKGEN's **PCLKMODE** with '1' (enable). (**RESETREG[0].[10]** release need.)
- 2) Set the **TIEOFFREG[3].[0]** with '1'.
- 3) Release resets of **RESETREG[0].[13]** and **RESETREG[0].[17]** (HDMI PHY reset release).
- 4) Set the HDMI PHY's registers with Table 37-3 to generate a pixel clock
- 5) Check the **PHY_READY** bit of the **PHY_STATUS_0** register (HDMI Link) whether the HDMI PHY's **PHY_READY** is HIGH.

Register address	Pixel clock frequency													
	25.2	25.17 5	27	27.02 7	54	54.05 4	74.25	74.17 6	148.5	148.35 2	25	65	108	162
0xC0100404	52h	D1h	D1h	D1h	51h	D1h	D1h	D1h	D1h	D1h	D1h	D1h	D1h	D1h
0xC0100408	3Fh	1Fh	22h	2Dh	2Dh	2Dh	1Fh	1Fh	1Fh	1Fh	27h	2Eh	1Fh	27h
0xC010040C	55h	50h	51h	72h	35h	32h	10h	10h	00h	00h	11h	12h	10h	14h
0xC0100410	40h	40h	40h	40h	40h	40h	40h	40h	40h	51h	61h	40h	51h	51h
0xC0100414	01h	20h	08h	64h	01h	64h	40h	5Bh	40h	5Bh	40h	40h	5Bh	5Bh
0xC0100418	00h	1Eh	FCh	12h	00h	12h	F8h	EFh	F8h	EFh	D6h	34h	EFh	A7h
0xC010041C	C8h	C8h	E0h	C8h	C8h	C8h	C8h	C8h	C8h	C8h	C8h	C8h	C8h	C8h
0xC0100420	82h	81h	98h	43h	82h	43h	81h	81h	81h	81h	82h	81h	84h	
0xC0100424	C8h	E8h	E8h	E8h	C8h	E8h	E8h	E8h	E8h	E8h	E8h	E8h	E8h	

0xC0100428	BDh	BDh	CBh	0Eh	0Eh	0Eh	BAh	B9h	BAh	B9h	E8h	16h	B9h	E8h
0xC010042C	D8h	D8h	D8h	D9h	D9h	D9h	D8h	D8h	D8h	D8h	D8h	D9h	D8h	D8h
0xC0100430	45h	45h	45h	45h	45h	45h	45h	45h	45h	45h	45h	45h	45h	45h
0xC0100434	A0h	A0h	A0h	A0h	A0h	A0h	A0h	A0h	A0h	A0h	A0h	A0h	A0h	A0h
0xC0100438	ACh	ACh	ACh	ACh	ACh	ACh	ACh	ACh	ACh	ACh	ACh	ACh	ACh	ACh
0xC010043C	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h
0xC0100440	06h	06h	06h	06h	06h	06h	56h	56h	66h	66h	56h	56h	56h	56h
0xC0100444	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h
0xC0100448	01h	09h	09h	09h	09h	09h	09h	09h	09h	09h	09h	09h	09h	09h
0xC010044C	84h	84h	84h	84h	84h	84h	84h	84h	84h	84h	84h	84h	84h	84h
0xC0100450	05h	05h	05h	05h	05h	05h	05h	05h	05h	05h	05h	05h	05h	05h
0xC0100454	22h	22h	22h	22h	22h	22h	22h	22h	22h	22h	22h	22h	22h	22h
0xC0100458	24h	24h	24h	24h	24h	24h	24h	24h	24h	24h	24h	24h	24h	24h
0xC010045C	86h	86h	86h	86h	86h	86h	86h	86h	86h	86h	86h	86h	86h	86h
0xC0100460	54h	54h	54h	54h	54h	54h	54h	54h	54h	54h	54h	54h	54h	54h
0xC0100464	F4h	F4h	E4h	E3h	E4h	E3h	A5h	A6h	4Bh	4Bh	84h	B9h	A6h	85h
0xC0100468	24h	24h	24h	24h	24h	24h	24h	24h	25h	25h	24h	25h	24h	24h
0xC010046C	00h	00h	00h	00h	01h	01h	01h	01h	03h	03h	01h	03h	01h	01h
0xC0100470	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h
0xC0100474	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h
0xC0100478	01h	01h	01h	01h	01h	01h	01h	01h	01h	01h	01h	01h	01h	01h
0xC010047C	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h	80h
0xC010048C	10h	10h	10h	10h	10h	10h	10h	10h	10h	10h	10h	10h	10h	10h

Table 37-3. HDMI PHY Configuration Table (8bit pixel)

37.4.4 HDMI PHY Register Summary

HDMI PHY's register has own mean. Following is for user reference.

Bit	R/W	Symbol	Description				Reset Value				
HDMIPHY 4h register											
<i>Address :C010 0404h</i>											
[31:8]	R	RESERVED	Reserved				24'b0				
[5:4]	R/W	CLK_SEL	Select the HDMI PHY reference clock Users must set both of TIEOFF's register and this Register. This register must be 0.								
HDMIPHY 24h register											
<i>Address :C010 0424h</i>											
[31:8]	R	RESERVED	Reserved				24'b0				
[7]	R/W	REF_CKO_SEL	0: REF_OCS 1 : Internal Reference Clock								
HDMIPHY 3ch register											
<i>Address :C010 043Ch</i>											
[31:8]	R	RESERVED	Reserved				24'b0				
[7]	R/W	TX_AMP_LVL[0]	TX_AMP_LVL[0] bit								
[5:4]	R/W	TX_RES[1:0]	TMDS Data Source Termination Resistor Control 0 : Source Termination OFF 1 : 300 Ohm								

Bit	R/W	Symbol	Description	Reset Value
			2 : 150 Ohm 3 : 100 Ohm	
HDMIPHY 40h register				
<i>Address :C010 0440h</i>				
[31:8]	R	RESERVED	Reserved	24'b0
[3:0]	R/W	TX_AMP_LVL[4:1]	TX_AMP_LVL[4:1] bit TMDS Data Amplitude Control. 1LSB corresponds to 50mVdiff amplitude level. 00000 : 400mVdiff (Min Value) 11111 : 1950 mVdiff (Max Value)	
HDMIPHY 5ch register				
<i>Address :C010 045Ch</i>				
[31:8]	R	RESERVED	Reserved	24'b0
[7:3]	R/W	TX_CLK_LVL[4:0]	TMDS Clock Amplitude Control 1LSB corresponds to 50mVdiff amplitude level. 00000 : 400mVdiff (Min Value) 11111 : 1950mVdiff (Max Value)	
HDMIPHY 74h register				
<i>Address :C010 0474h</i>				
[31:8]	R	RESERVED	Reserved	24'b0
[7]	R/W	APB_PDEN	If APB_PDEN = 1, power down of each building blocks of PHY can be controlled APB Reg74 bit<6:4>, bit<2:0> 0 : Disable 1 : Enable	
[6]	R/W	PLL_PD	0 : Normal Status 1 : Power Down Status PLL & Bias Block Power Down	
[5]	R/W	TX_CLKSER_PD	Clock Serializer Power Down	
[4]	R/W	TX_CLKDRV_PD	TMDS Clock Driver Power Down	
[2]	R/W	TX_DRV_PD	TMDS Data Driver Power Down	
[1]	R/W	TX_SER_PD	TMDS Data Serializer Power Down	
[0]	R/W	TX_CLK_PD	TX Internal Clock Buffer / Divider Power Down	
HDMIPHY 78h register				
<i>Address :C010 0478h</i>				
[31:8]	R	RESERVED	Reserved	24'b0
[7]	R/W	TESTEN	PHY Test Mode Enable 0 : Normal Operation Mode 1 : PHY Test Mode	
[6:0]	R/W	TEST	PHY Test Mode Control Signal	
HDMIPHY 7Ch register				
<i>Address :C010 047Ch</i>				
[31:8]	R	RESERVED	Reserved	24'b0
[7]	R/W	MODE_SET_DONE	An indicator of APB setting state. Refer to the Section 1.5.1 PHY Configuration Change Through APB	

37.5 HDMI Application Sequences

Users must be set to the following sequence in order to user HDMI.

- HDMI PHY configuration
- I2S (or SPDIFTX) configuration for the source audio data
- DPC (or Resolution Converter) configuration for the source video data
- HDMI Link configuration
- HDMI Converter configuration

37.6 DisplayTop Register Summary

User use this register to select the RGB Video data between the Primary DPC and the Secondary DPC.

Bit	R/W	Symbol	Description	Reset Value
DISPLAYTOP HDMI MUX Control register (HDMI_MUXCTRL)				
<i>Address :C010 1004h</i>				
[31]	R/W	HDMI_MUXENB	MUX Enable 0: MUX Disable 1 : MUX Enable	1'b0
[30:2]	R/W	RESERVED	Reserved	29'b0
[1:0]	R/W	HDMI_MUXSEL	MUX Select 0: Primary DPC 1 : Secondary DPC 2-3 : Reserved (Never use this value)	2'b0
DISPLAYTOP HDMI sync Control register 0 (HDMI_syncctrl0)				
<i>Address :C010 1014h</i>				
[31]	R/W	HDMI_VCLK_SEL	Must set this value to 0 0: HDMI PHY's pixel clock used for HDMI Operation 1 : Never set this value	1'b0
[30:16]	R/W	RESERVED	Reserved	15b0
[15:0]	R/W	HDMI_VSYNCSTART	Specifies the start line of i_v_sync for the HDMI Link. Refer Table 1-1	16'b0
DISPLAYTOP HDMI sync Control register 1 (HDMI_syncctrl1)				
<i>Address :C010 1018h</i>				
[31:16]	R/W	RESERVED	Reserved	16'b0
[15:0]	R/W	HDMI_HACTIVESTART	Specifies the start position(h_line) of h_active for the HDMI Link. Refer Table 1-1	16'b0
DISPLAYTOP HDMI sync Control register 2 (HDMI_syncctrl2)				
<i>Address :C010 1018h</i>				
[31:16]	R/W	RESERVED	Reserved	16'b0
[15:0]	R/W	HDMI_HACTIVEEND	Specifies the end position of h_active for the HDMI Link. Refer Table 1-1	16'b0
DISPLAYTOP HDMI sync Control register 3 (HDMI_syncctrl3)				
<i>Address :C010 1018h</i>				
[31:16]	R/W	HDMI_VSYNCHSEND	Specifies the end position of i_v_sync for the HDMI Link. Refer Table 1-1	16'b0
[15:0]	R/W	HDMI_VSYNCHSTART	Specifies the start position of i_v_sync for the HDMI Link. Refer Table 1-1	16'b0

Section 38. MIPI

38.1 Overview

The NXP4330D/Q has a MIPI-DSI master and a MIPI-CSI slave.

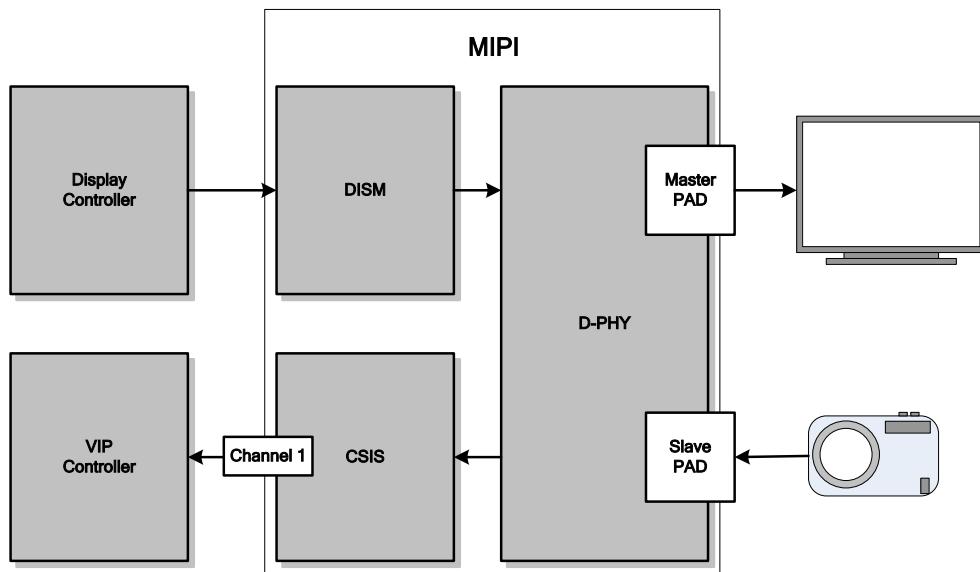


Figure 38-1. MIPI-DSI and MIPI-CSI

38.1.1 Features

DSI master features (DSIM)

The key features of MIPI DSIM include:

- MIPI DSI Standard Specification V1.01r11
- Maximum resolution ranges up to WUXGA (1920x1200)
- Supports 1, 2, 3, or 4 data lanes
- Supports pixel format: 16bpp, 18bpp packed, 18bpp loosely packed (3 byte format), and 24bpp
- Interfaces
 - Supports RGB Interface for Video Image Input from display controller
 - Supports PMS control interface for PLL to configure byte clock frequency
 - Supports Prescaler to generate escape clock from byte clock

CSI slave features (CSIS)

- Support YUV422 of 8-bits only. See VIP for more information.

D-PHY features

The features of MIPI D-PHY are:

- The maximum high speed clock frequency of MIPI D-PHY core is 1 GHz.
- D-PHY spec v1.00 compatible.
- Synchronous link between Master (data source) and Slave (data sink).
- All lanes support high-speed transmission in forward direction.
- Bi-directional data transmission in Low-Power mode at the Master Data Lane 0 only.
- Use token passing to control the communication direction of the link.
- High-Speed mode for fast data traffics and Low-Power mode for controls and low speed data transmission.
- High-Speed mode: differential and terminated, 200 mV swing: 80 to 1000 Mbps
- Low-Power mode: single-ended and non-terminated, 1.2 V swing: 10 Mbps maximum
(Use this mode for low-speed asynchronous data communications or controls)

38.2 DSIM

38.2.1 Block Diagram of MIPI DSI System

Total System Block Diagram

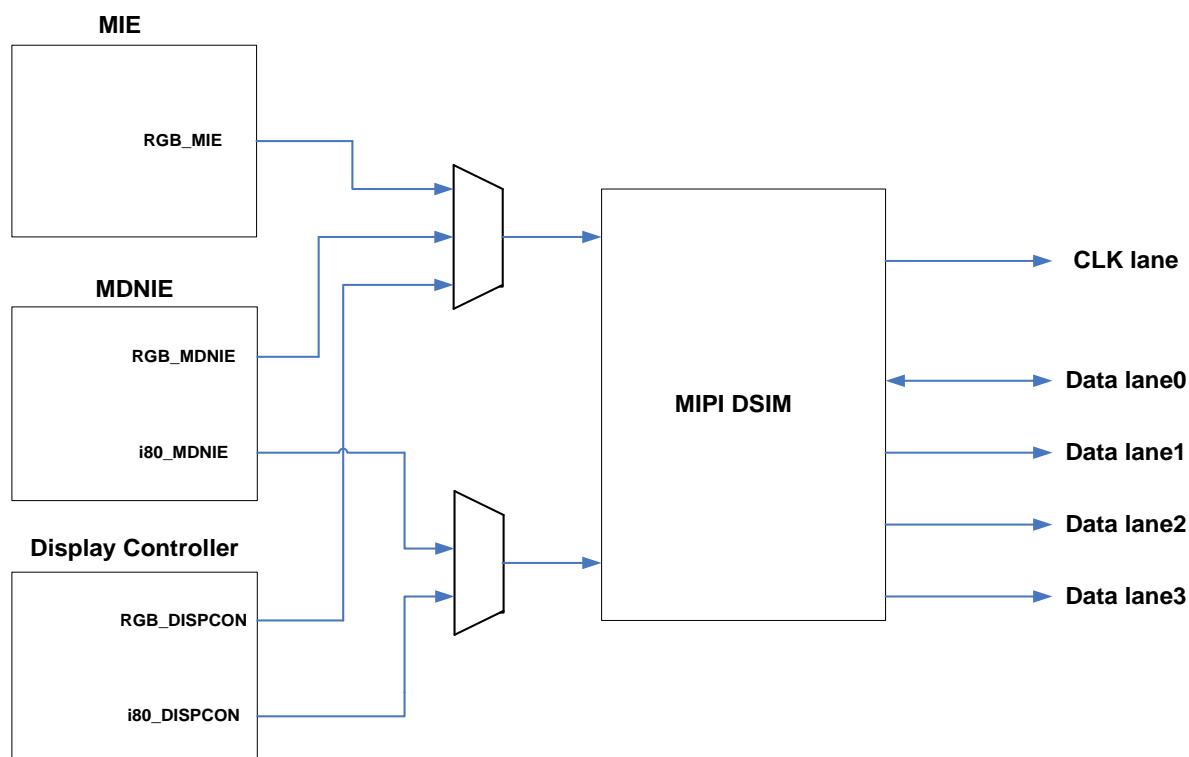


Figure 38-2 MIPI DSI System Block Diagram

- DSIM gets data from the three different IPs, namely, MIE, MDNIE, and Display Controller.
- You can select one of above data paths by setting MDNIE registers.

Internal Primary FIFOs

Table 38-1 describes configurable-sized primary FIFOs.

Port	FIFO Type	Size	Description
Main display	Packet Header FIFO	3byte X 128 depth	Specifies the packet header FIFO for main display.
	Payload FIFO	4byte X 2048 depth	Specifies the payload FIFO for main display image.
Sub display for I80 INTERFACE image data	Packet Header FIFO	3byte X 4 depth	Specifies the packet header FIFO for I80 INTERFACE sub display.
	Payload FIFO	4byte X 512 depth	Specifies the payload FIFO for I80 INTERFACE sub display

			image.
Command for I80 INTERFACE command	Packet Header FIFO	3byte X 16 depth	Specifies the packet header FIFO for I80 INTERFACE command packet.
	Payload FIFO	4byte X 16 depth	Specifies the payload FIFO for I80 INTERFACE command long packet payload.
SFR for general packets	Packet Header FIFO	3byte X 16 depth	Specifies the packet header FIFO for general packet.
	Payload FIFO	4byte X 512 depth	Specifies the payload FIFO for general long packet.
RX FIFO	Packet header and Payload FIFO	4byte X 64 depth	Specifies Rx FIFO for LPDR. This FIFO is common for packet header and payload.

Table 38-1. Internal Primary FIFO List

Packet Header Arbitration

There are four-packet headers FIFOs for Tx, namely, main display, sub display, I80 INTERFACE command, and SFR FIFO. The main and sub display FIFO packet headers contain the image data, while the I80 INTERFACE command FIFO packet header contains the command packets. On the other hand, the SFR FIFO packet header contains command packets, sub display image data (in Video mode), and so on.

The packet header arbiter has a “Fixed priority” algorithm. Priority order is fixed as main display, sub display, I80 INTERFACE command, and SFR FIFO packet header.

In the Video mode, sub display and I80 INTERFACE command FIFO are not used. The SFR FIFO packet header checks if the main display FIFO is empty (no request) in not-active image region and then sends its request.

RxFIFO Structure

To read the packets received via low power data receiving mode, RxFIFO acts like an SFR. RxFIFO is an asynchronous FIFO with ByteClk and PCLK domains as input clock and output clock domains respectively. The Rx data is synchronized to RxClk. RXBUF has four Rx Byte buffers for aligning byte to word.

The packet headers of all the packets stored in RxFIFO are word-aligned, that is, the first byte of a packet is always stored in LSB. For example, if a long packet has 7-byte payload, the last byte is filled with dummy byte and the next packet is stored in the next word, as shown in Figure 38-3.

NOTE: CRC data is not stored in RxFIFO.

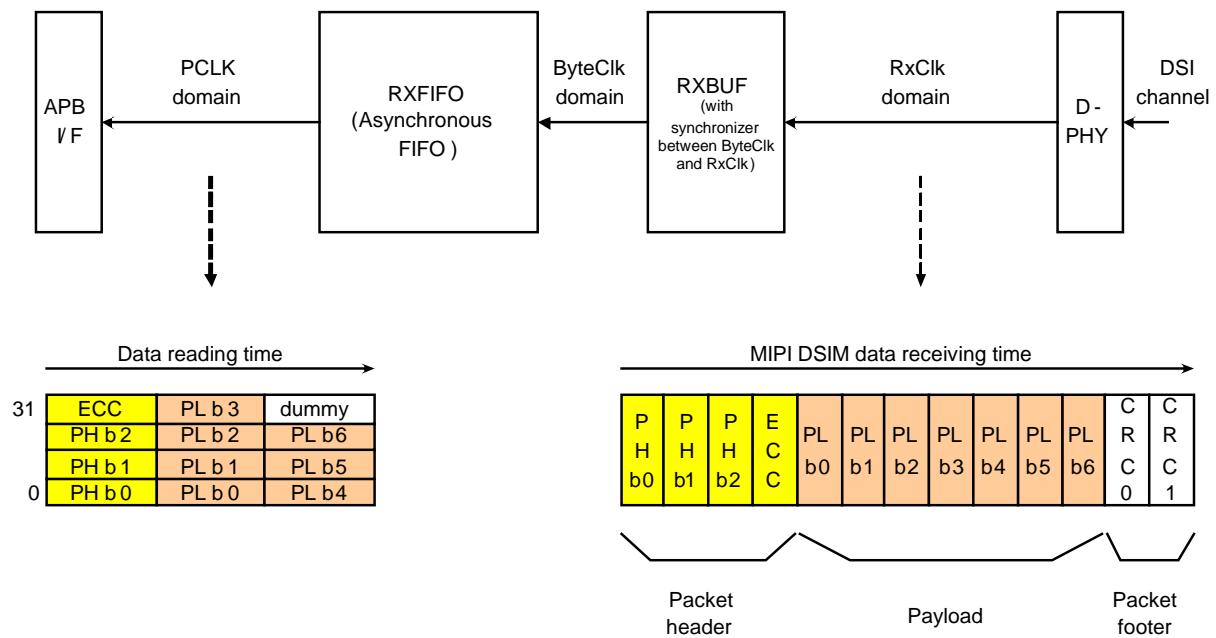


Figure 38-3 Rx Data Word Alignment

38.2.2 Interfaces and Protocol

Display Controller-to-DSI Conceptual Signal Converting Diagram in Video Mode

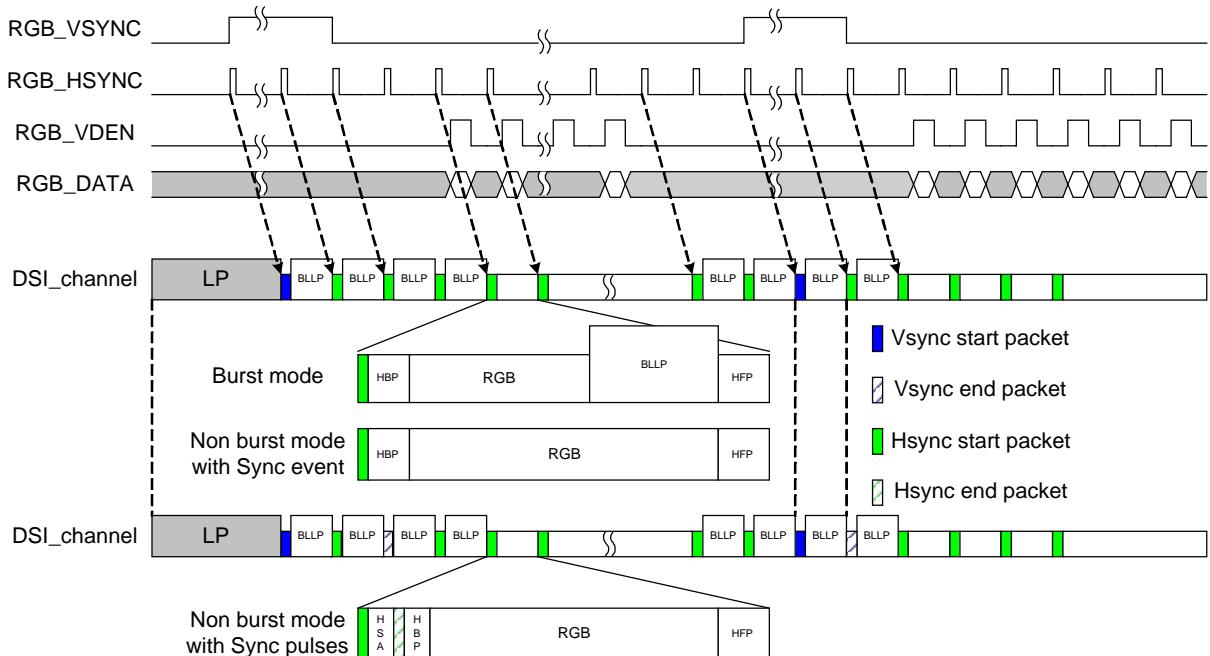


Figure 38-4 Signal Converting Diagram in Video Mode

Display Controller Interface

MIPI DSI Master has two-display controller interfaces, namely, RGB INTERFACE for main display and CPU INTERFACE (I80 INTERFACE) for main/ sub display. The Video mode uses RGB INTERFACE while the Command mode uses CPU INTERFACE.

The RGB image data is loaded on the data bus of RGB INTERFACE and I80 INTERFACE with the same order: RGB_VD[23:0] or SYS_VDOUT[23:0] is {R[7:0],G[7:0],B[7:0]}. Each byte aligns to the most significant bit. For instance, in the 12-bit mode, only three 4-bit values are valid as R, G, and B each, that is, data[23:20], data[15:12], and data[7:4]. The DSIM ignores rest of the bits.

RGB Interface

Vsync, Hsync, and VDEN are active high signals. Among the three signals, Vsync and Hsync are pulse types that spend several video clocks. RGB_VD[23:0] is {R[7:0],G[7:0],B[7:0]}. All sync signals are synchronized to the rising edge of RGB_VCLK. The display controller sends minimum one horizontal line length of Vsync pulse, V back porch, and V front porch. Hsync pulse width should be longer than 1-byte clock cycle.

HSA Mode

HSA mode specifies the Horizontal Sync Pulse area disable mode.

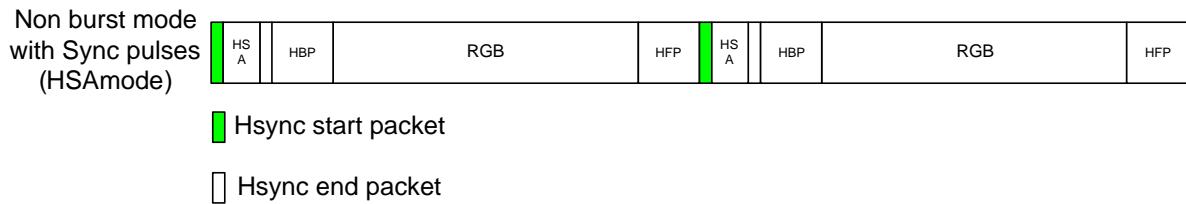


Figure 38-5 Block Timing Diagram of HSA Mode (HSA mode reset: DSIM_CONFIG[20] = 0)

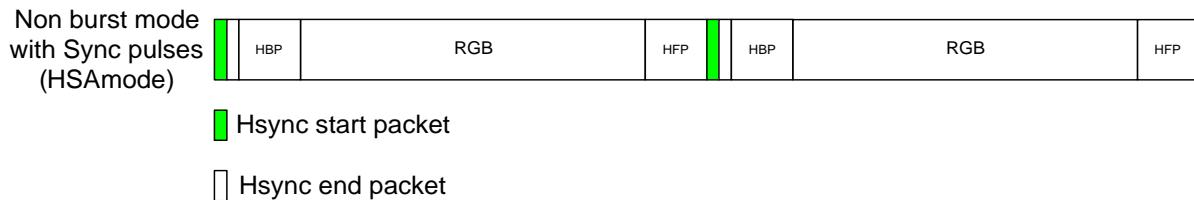


Figure 38-6 Block Timing Diagram of HSA Mode (HSA mode set: DSIM_CONFIG[20] = 1)

HBP mode HBP mode specifies the Horizontal Back Porch disable mode.

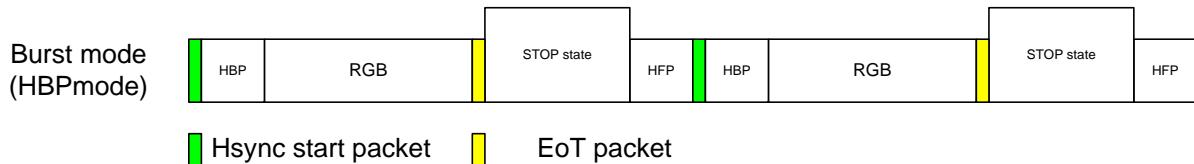


Figure 38-7 Block Timing Diagram of HBP Mode (HBP Mode Reset: DSIM_CONFIG[21] = 0)

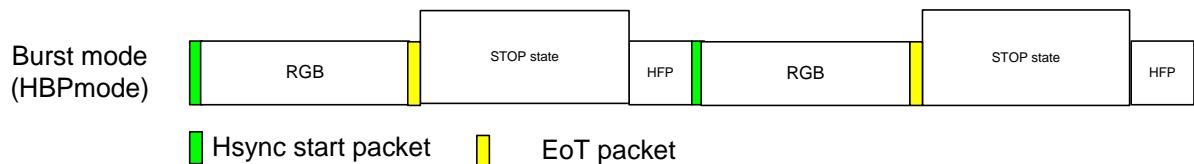


Figure 38-8 Block Timing Diagram of HBP Mode (HBP Mode Set: DSIM_CONFIG[21] = 1)

HFP mode HFP mode specifies the Horizontal Front Porch disable mode.

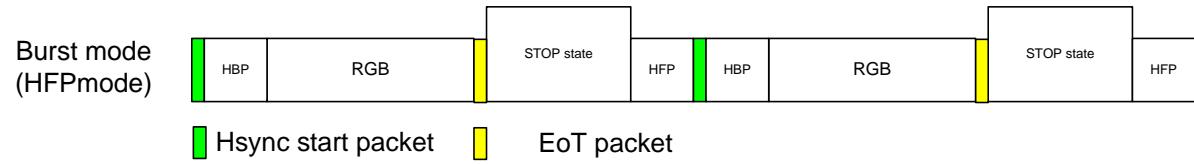


Figure 38-9 Block Timing Diagram of HFP Mode (HFP Mode Reset: DSIM_CONFIG[22] = 0)

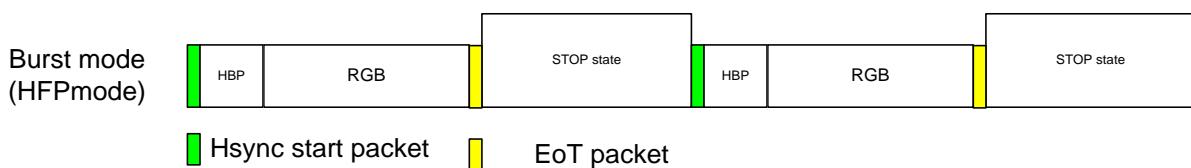


Figure 38-10 Block Timing Diagram of HFP Mode (HFP Mode Set: DSIM_CONFIG[22] = 1)

HSE Mode

HSE mode specifies the Horizontal Sync End Packet Enable mode in Vsync pulse or Vporch area.

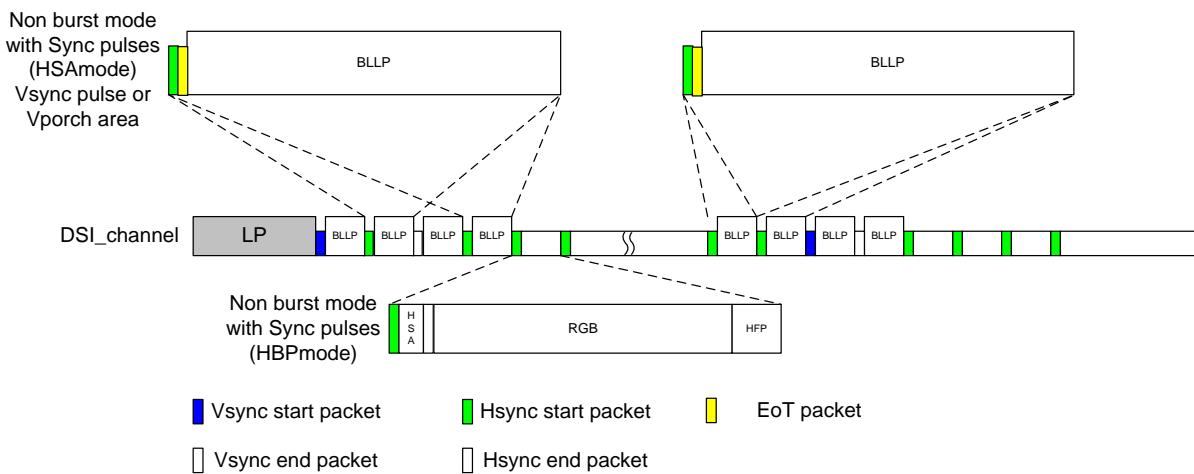


Figure 38-11 Block Timing Diagram of HSE Mode (HSE Mode Reset: DSIM_CONFIG[23] = 0)

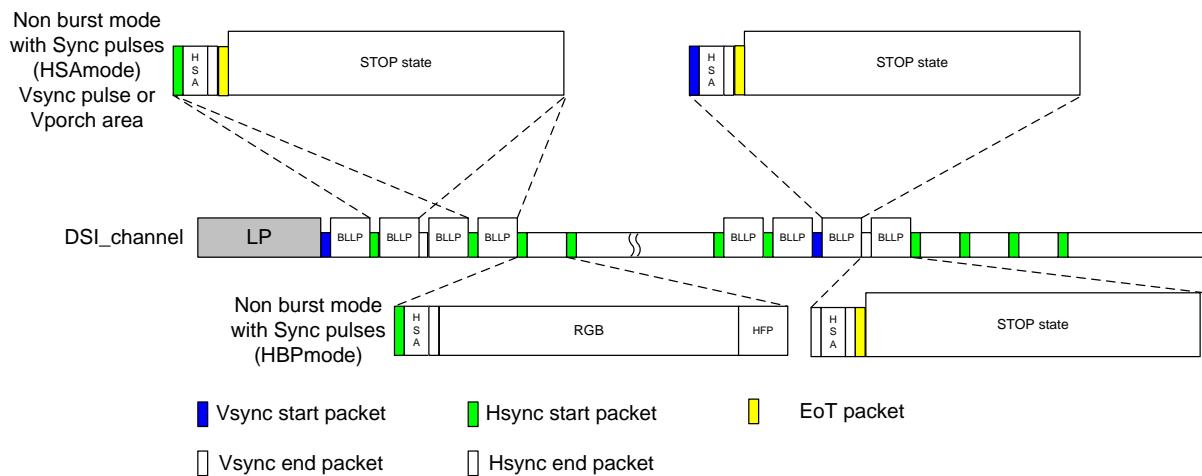


Figure 38-12 Block Timing Diagram of HSE Mode (HSE Mode Set: DSIM_CONFIG[23] = 1)

Transfer General Data in Video Mode

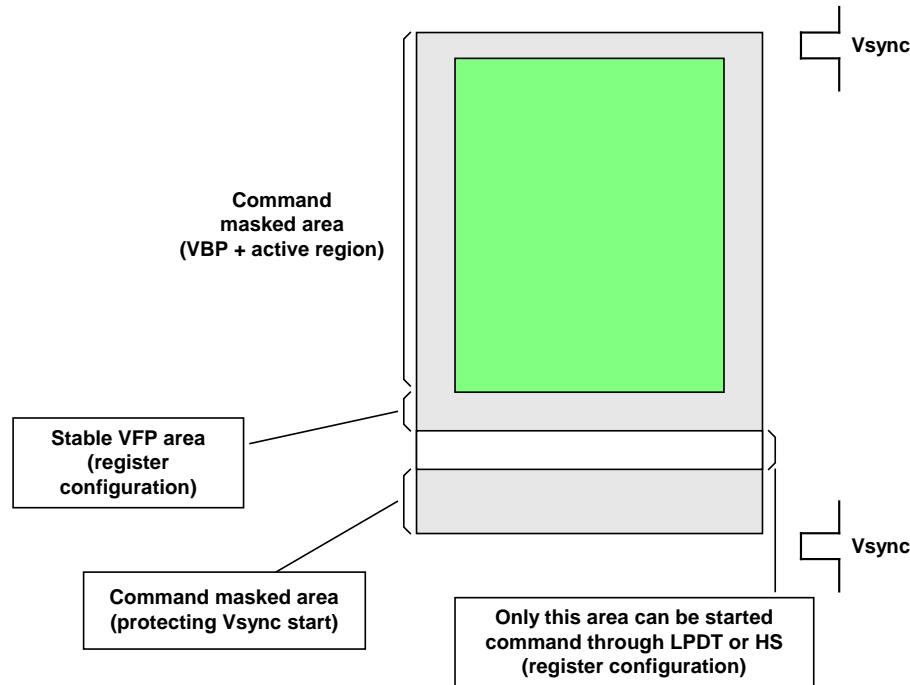


Figure 38-13 Stable VFP Area Before Command Transfer Allowing Area

MIPI DSIM Converts RGB Interface to Video Mode

Vsync and Hsync packets are extremely important to protect image in Video mode. MIPI DSIM allows several lines in VFP area to transfer general data transfer. As shown in Figure 38-13, the vertical front porch is divided into three areas, namely, stable VFP area, command allowed area, and command masked area.

The register configures stable VFP area. Configuration boundary is 11'h000 ~ 11'h7FFF in DSIM_MVPORCH.

The register also configures the command allowed area. Configuration boundary is 4'h0 ~ 4'hF in DSIM_MVPORCH. Only this area is allowed to start "command transfer" through HS mode or LPDT. In LPDT, data transferring takes a long time to complete (approximately hundreds of microseconds or more). In this time, Hsync packet does not arrive due to LPDT long packet. MIPI DSIM comprises of big size FIFO for lost Hsync packet. After LPDT, MIPI DSIM transfers these Hsync packets immediately through HS mode.

To protect Vsync, command masked area is masked area. This area is calculated using LPDT bandwidth. For example, if EscClk is 10MHz, the maximum long packet payload size is 1KB and LPDT, LPDT transferring time is 824us (packet size: 1030byte, LPDT maximum bandwidth: 10Mbps). If one line time is 20us, the line timing violation occurs in 42 lines. Therefore, command masked area is larger than $42 + \alpha$. This ' α ' is transferring time of the violated Hsync packets.

Display controller should be configured in such a way that VFP lines are sum of stable vfp, command allowed area, and command masked area.

Relation between Input Transactions and DSI Transactions

Input Interface	Input transaction	DSI Transaction
RGB	RGB transaction	Specifies the RGB Packet. 888, 666, 666 (loosely packed), and 565 should be specified via register configuration.

I80	I80 Image Transaction	Specifies the Data type, that is, "DCS Long Write packet". (DCS command is "memory write start/continue".)
I80	I80 Command Transaction	Specifies any DSI packet. Bytes in I80 transaction should be the same bytes in DSI packets.
SFR	Header and Payload FIFO access	Specifies any DSI Packets. Bytes in APB transaction should be the same bytes in DSI packets.

Table 38-2. Relation Between Input Transactions and DSI Transactions

38.2.3 Configuration

Video Mode Versus Command Mode

MIPI DSI Master Block supports two modes, namely, Video mode and Command mode.

38.2.4 PLL

To transmit Image data, MIPI DSI Master Block needs high frequency clock (80MHz ~ 1GHz) generated by PLL.

To configure PLL, MIPI DSI Master comprises of SFRs and corresponding interface signals. PLL is embedded in PHY module. You should use other PLL in SoC if it meets the timing specification.

38.2.5 Buffer

In MIPI DSI standard specification, DSI Master sends image stream in burst mode. The image stream transmits in high-speed and bit-clock frequency. This mode allows the device to stay in stop state longer to reduce power consumption. For this mode, MIPI DSI Master has a dual line buffer to store one complete line and send it faster at the next line time.

38.2.6 DSIM Register Summary

DSIM_BASE address : 0xC00D0100

Register Map

Register	Address	R/W	Description	Reset Value
DSIM_STATUS	DSIM_BASE + 0x0000	R	Specifies the status register.	0x0010_010F
DSIM_SWRST	DSIM_BASE + 0x0004	R/W	Specifies the software reset register.	0x0000_0000
DSIM_CLKCTRL	DSIM_BASE + 0x0008	R/W	Specifies the clock control register.	0x0000_FFFF
DSIM_TIMEOUT	DSIM_BASE + 0x000C	R/W	Specifies the time out register.	0x0FF_FFFF
DSIM_CONFIG	DSIM_BASE + 0x0010	R/W	Specifies the configuration register.	0x0200_0000
DSIM_ESCMODE	DSIM_BASE + 0x0014	R/W	Specifies the escape mode register.	0x0000_0000
DSIM_MDRESOL	DSIM_BASE + 0x0018	R/W	Specifies the main display image resolution register.	0x0300_0400
DSIM_MVPORCH	DSIM_BASE + 0x001C	R/W	Specifies the main display Vporch register.	0xF000_0000
DSIM_MHPORCH	DSIM_BASE + 0x0020	R/W	Specifies the main display Hporch register.	0x0000_0000
DSIM_MSYNC	DSIM_BASE + 0x0024	R/W	Specifies the main display Sync Area register.	0x0000_0000
DSIM_SDRESOL	DSIM_BASE + 0x0028	R/W	Specifies the sub display image resolution register.	0x0300_0400
DSIM_INTSRC	DSIM_BASE + 0x002C	R/W	Specifies the interrupt source register.	0x0000_0000
DSIM_INTMSK	DSIM_BASE + 0x0030	R/W	Specifies the interrupt mask register.	0xBB37_FFFF

DSIM_PKTHDR	DSIM_BASE + 0x0034	W	Specifies the packet header FIFO register.	0x0000_0000
DSIM_PAYLOAD	DSIM_BASE + 0x0038	W	Specifies the payload FIFO register.	0x0000_0000
DSIM_RXFIFO	DSIM_BASE + 0x003C	R	Specifies the read FIFO register.	0xFFFF_FFFF
DSIM_FIFOTHLD	DSIM_BASE + 0x0040	R/W	Specifies the FIFO threshold level register.	0x0000_01FF
DSIM_FIFOCTRL	DSIM_BASE + 0x0044	R/W	Specifies the FIFO status and control register.	0x0155_551F
DSIM_MEMACCHR	DSIM_BASE + 0x0048	R/W	Specifies the FIFO memory AC characteristic register.	0x0000_4040
DSIM_PLLCTRL	DSIM_BASE + 0x004C	R/W	Specifies the PLL control register.	0x0000_0000
DSIM_PLLCTRL1	DSIM_BASE + 0x0050	R/W	Specifies the PLL control register 1.	0x0000_0000
DSIM_PLLCTRL2	DSIM_BASE + 0x0054	R/W	Specifies the PLL control register 2.	0x0000_0000
DSIM_PLLTMR	DSIM_BASE + 0x0058	R/W	Specifies the PLL timer register.	0xFFFF_FFFF
DSIM_PHYCTRL	DSIM_BASE + 0x005C	R/W	Specifies the D-PHY control register	0x0000_0000
DSIM_PHYCTRL1	DSIM_BASE + 0x0060	R/W	Specifies the D-PHY control register 1	0x0000_0000
DSIM_PHYTIMING	DSIM_BASE + 0x0064	R/W	Specifies the D-PHY timing register	0x0000_0000
DSIM_PHYTIMING1	DSIM_BASE + 0x0068	R/W	Specifies the D-PHY timing register 1	0x0000_0000
DSIM_PHYTIMING2	DSIM_BASE + 0x006C	R/W	Specifies the D-PHY timing register 2	0x0000_0000
DSIM_VERSION	DSIM_BASE + 0x0070	R	Specifies the DSIM version register	0x8000_0001

Status Register (DSIM_STATUS, R, Address = DSIM_BASE + 0x0000)

This register reads and checks internal and interface status. It also checks FSM status, Line buffer status, current image line number, and so on.

DSIM_STATUS	Bit	Description	Initial State
PllStable	[31]	D-phy pll generates stable byteclk.	0
Reserved	[30:21]	Reserved	0
SwRstRs	[20]	Specifies the software reset status. 0 = Reset state 1 = Release state	1
Reserved	[19:17]	Reserved	0
Direction	[16]	Specifies the data direction indicator. 0 = Forward direction 1 = Backward direction	0
Reserved	[15:11]	Reserved	0
TxReadyHsClk	[10]	Specifies the HS clock ready at clock lane. 0 = Not ready for transmitting HS data at clock lane 1 = Ready for transmitting HS data at clock lane	0
UlpsClk	[9]	Specifies the ULPS indicator at clock lane. 0 = No ULPS in clock lane 1 = ULSP in clock lane	0
StopstateClk	[8]	Specifies the stop state indicator at clock lane. 0 = No stop state in clock lane 1 = Stop state in clock lane	1
UlpsDat[3:0]	[7:4]	Specifies the ULPS indicator at data lanes. UlpsDat[0]: Data lane 0 UlpsDat[1]: Data lane 1 UlpsDat[2]: Data lane 2	0

		UlpsDat[3]: Data lane 3 0 = No ULPS in each data lane 1 = ULPS in each data lane	
StopstateDat[3:0]	[3:0]	<p>Specifies the stop state indicator at data lane.</p> <p>StopstateDat[0]: Data lane 0 StopstateDat[1]: Data lane 1 StopstateDat[2]: Data lane 2 StopstateDat[3]: Data lane 3</p> <p>0 = No stop state in each data lane 1 = Stop state in each data lane</p>	0xF

Software Reset Register (DSIM_SWRST, R/W, Address = DSIM_BASE + 0x0004)

DSIM_SWRST	Bit	Description	Initial State
Reserved	[31:17]	Reserved	-
FuncRst	[16]	<p>Specifies the software reset (High active).</p> <p>"Software reset" resets all FF in MIPI DSIM (except SFRs: STATUS, SWRST, CLKCTRL, TIMEOUT, CONFIG, ESCMODE*, MDRESOL, MDVPORCH, MHPORCH, MSYNC, INTMSK, SDRESOL, FIFOFLD, FIFOCTRL**, MEMACCHR, PLLCTRL, PLLTMR, PHYACCHR, and VERINFORM).</p> <p>0 = Standby 1 = Reset</p> <p>*: ForceStopstate, CmdLpd, TxLpd, **: nInitRx, nInitSfr, nInitB0, nInitSub, nInitMD</p>	0
Reserved	[15:1]	Reserved	-
SwRst	[0]	<p>Specifies the software reset (High active).</p> <p>"Software reset" resets all FF in MIPI DSIM (except some SFRs: STATUS, SWRST, CLKCTRL, PLLCTRL, PLLTMR, and PHYTUNE).</p> <p>0 = Standby 1 = Reset</p>	0

Clock Control Register (DSIM_CLKCTRL, R/W, Address = DSIM_BASE + 0x0008)

DSIM_CLKCTRL	Bit	Description	Initial State
TxRequestHsClk	[31]	Specifies the HS clock request for HS transfer at clock lane (Turn on HS clock)	0
Reserved	[30:29]	Reserved	-
EscClkEn	[28]	Enables the escape clock generating prescaler.	0
		0 = Disables 1 = Enables	
PLLbypass	[27]	Sets the PLLbypass signal connected to D-PHY module input for selecting clock source bit. 0 = PLL output 1 = External Serial clock This bit must be set to 0.	0
ByteClkSrc	[26:25]	Selects byte clock source. (It must be 00.) 00 = D-PHY PLL (default). PLL_out clock is used to generate ByteClk by dividing 8.	0
ByteClkEn	[24]	Enables byte clock.	0
		0 = Disables 1 = Enables	

DSIM_CLKCTRL	Bit	Description	Initial State
LaneEscClkEn	[23:19]	<p>Enables escape clock for D-phy lane.</p> <p>LaneEscClkEn[0] = Clock lane LaneEscClkEn[1] = Data lane 0 LaneEscClkEn[2] = Data lane 1 LaneEscClkEn[3] = Data lane 2 LaneEscClkEn[4] = Data lane 3</p> <p>0 = Disables 1 = Enables</p>	0
Reserved	[18:16]	Reserved	-
EscPrescaler	[15:0]	<p>Specifies the escape clock prescaler value.</p> <p>The escape clock frequency range varies up to 20MHz.</p> <p>Note: The requirement for BTA is that the Host Escclk frequency should range between 66.7 ~ 150% of the peripheral escape clock frequency.</p> <p>EscClk = ByteClk / (EscPrescaler)</p>	0xFFFF

Time Out register (DSIM_TIMEOUT, R/W, Address = DSIM_BASE + 0x000C)

DSIM_TIMEOUT	Bit	Description	Initial State
Reserved	[31:24]	Reserved	-
BtaTout	[23:16]	<p>Specifies the timer for BTA.</p> <p>This register specifies time out from BTA request to change the direction with respect to Tx escape clock.</p>	0xFF
LpdrTout	[15:0]	<p>Specifies the timer for LP Rx mode timeout. This register specifies time out on how long RxValid deasserts, after RxLpd asserts with respect to Tx escape clock.</p> <p>RxValid specifies Rx data valid indicator.</p> <p>RxLpd specifies an indicator that D-phy is under RxLpd mode.</p> <p>RxValid and RxLpd specifies signal from D-phy.</p>	0xFFFF

Configuration Register (DSIM_CONFIG, R/W, DSIM_BASE + 0x0010)

This register configures MIPI DSI master such as data lane number, input interface, porch area, frame rate, BTA, LPDT, ULPS, and so on.

DSIM_CONFIG	Bit	Description	Initial State
Reserved	[31:30]	Reserved	-
Mflush_VS	[29]	<p>Auto flush of MD FIFO using Vsync pulse.</p> <p>It needs that Main display FIFO should be flushed for deleting garbage data.</p> <p>0 = Enable (default) 1 = Disable</p>	0
EoT_r03	[28]	<p>Disables EoT packet in HS mode.</p> <p>0 = Enables EoT packet generation for V1.01r11 1 = Disables EoT packet generation for V1.01r03</p>	0
Synchinform	[27]	<p>Selects Sync Pulse or Event mode in Video mode.</p> <p>0 = Event mode (non burst, burst) 1 = Pulse mode (non burst only)</p> <p>In command mode, this bit is ignored.</p>	0

DSIM_CONFIG	Bit	Description	Initial State
BurstMode	[26]	<p>Selects Burst mode in Video mode</p> <p>In Non-burst mode, RGB data area is filled with RGB data and Null packets, according to input bandwidth of RGB interface.</p> <p>In Burst mode, RGB data area is filled with RGB data only.</p> <p>0 = Non-burst mode 1 = Burst mode</p> <p>In command mode, this bit is ignored.</p>	0
VideoMode	[25]	<p>Specifies display configuration.</p> <p>0 = Command mode 1 = Video mode</p>	1
AutoMode	[24]	<p>Specifies auto vertical count mode.</p> <p>In Video mode, the vertical line transition uses line counter configured by VSA, VBP, and Vertical resolution. If this bit is set to '1', the line counter does not use VSA and VBP registers.</p> <p>0 = Configuration mode 1 = Auto mode</p> <p>In command mode, this bit is ignored.</p>	0
HseMode	[23]	<p>In Vsync pulse and Vporch area, MIPI DSI master transfers only Hsync start packet to MIPI DSI slave at MIPI DSI spec 1.1r02. This bit transfers Hsync end packet in Vsync pulse and Vporch area (optional).</p> <p>0 = Disables transfer 1 = Enables transfer</p> <p>In command mode, this bit is ignored.</p>	0
HfpMode	[22]	<p>Specifies HFP disable mode. If this bit set, DSI master ignores HFP area in Video mode.</p> <p>0 = Enables 1 = Disables</p> <p>In command mode, this bit is ignored.</p>	0
HbpMode	[21]	<p>Specifies HBP disable mode. If this bit set, DSI master ignores HBP area in Video mode.</p> <p>0 = Enables 1 = Disables</p> <p>In command mode, this bit is ignored.</p>	0
HsaMode	[20]	<p>Specifies HSA disable mode. If this bit set, DSI master ignores HSA area in Video mode.</p> <p>0 = Enables 1 = Disables</p> <p>In command mode, this bit is ignored.</p>	0
MainVc	[19:18]	Specifies virtual channel number for main display.	0
SubVc	[17:16]	Specifies virtual channel number for sub display.	0
Reserved	[15]	Reserved	-
MainPixelFormat	[14:12]	<p>Specifies pixel stream format for main display.</p> <p>000 = 3bpp (for Command mode only) 001 = 8bpp (for Command mode only) 010 = 12bpp (for Command mode only) 011 = 16bpp (for Command mode only) 100 = 16-bit RGB (565) (for Video mode only) 101 = 18-bit RGB (666: packed pixel stream) (for Video mode only) 110 = 18-bit RGB (666: loosely packed pixel stream) for Common 111 = 24-bit RGB (888) for common</p>	0

DSIM_CONFIG	Bit	Description	Initial State
Reserved	[11]	Reserved	-
SubPixFormat	[10:8]	<p>Specifies pixel stream format for sub display.</p> <p>000 = 3bpp (for Command mode only) 001 = 8bpp (for Command mode only) 010 = 12bpp (for Command mode only) 011 = 16bpp (for Command mode only) 100 = 16-bit RGB (565) (for Video mode only) 101 = 18-bit RGB (666: packed pixel stream) for Video mode only 110 = 18-bit RGB (666: loosely packed pixel stream) for common 111 = 24-bit RGB (888) (for Common)</p>	0
Reserved	[7]	Reserved	-
NumOfDatLane	[6:5]	<p>Sets the data lane number.</p> <p>00 = Data lane 0 (1 data lane) 01 = Data lane 0 ~ 1 (2 data lanes) 10 = Data lane 0 ~ 2 (3 data lanes) 11 = Data lane 0 ~ 3 (4 data lanes)</p>	0
LaneEn[4:0]	[4:0]	<p>Enables the lane. If Lane_EN is disabled, the lane ignores input and drives initial value through output port.</p> <p>0 = Lane is off. 1 = Lane is on.</p> <p>LaneEn[0] = Clock lane enabler LaneEn[1] = Data lane 0 enabler LaneEn[2] = Data lane 1 enabler LaneEn[3] = Data lane 2 enabler LaneEn[4] = Data lane 3 enabler</p>	0

Escape Mode Register (DSIM_ESCMODE, R/W, Address = DSIM_BASE + 0x0014)

This register configures MIPI DSI master.

DSIM_ESCMODE	Bit	Description	Initial State
STOPstate_Cnt	[31:21]	<p>After transmitting read packet or write "set_tear_on" command, BTA requests to D-phy automatically. This counter value specifies the interval value between transmitting read packet (or write "set_tear_on" command) and BTA request.</p> <p>11'h000 = 2 EscClk 11'h001 = 2 EscClk + 1 EscClk ~ 11'h3FF = 2 EscClk + 1023 EscClk</p>	0
ForceStopstate	[20]	Forces Stopstate for D-PHY.	0
Reserved	[19:17]	Reserved	-
ForceBta	[16]	<p>Forces Bus Turn Around.</p> <p>1 = Sends the protocol layer request to D-PHY. MIPI DSI peripheral becomes master after BTA sequence. This bit clears automatically after receiving BTA acknowledge from MIPI DSI peripheral.</p>	0
Reserved	[15:8]	Reserved	-
CmdLpdt	[7]	Specifies LPDT transfers command in SFR FIFO.	0
		0 = HS Mode 1 = LP Mode	
TxLpdt	[6]	Specifies data transmission in LP mode (all data transfer in LPDT).	0
		0 = HS Mode 1 = LP Mode	
Reserved	[5]	Reserved	-

DSIM_ESCMODE	Bit	Description	Initial State
TxTriggerRst	[4]	Specifies remote reset trigger function. After trigger operation, these bits will be cleared automatically.	0
TxUlpsDat	[3]	Specifies ULPS request for data lane. Manually clears after ULPS exit.	0
TxUlpsExit	[2]	Specifies ULPS exit request for data lane. Manually clears after ULPS exit.	0
TxUlpsClk	[1]	Specifies ULPS request for clock lane. Manually clears after ULPS exit.	0
TxUlpsClkExit	[0]	Specifies ULPS exit request for clock lane. Manually clears after ULPS exit.	0

Main Display Image Resolution Register (DSIM_MDRESOL, R/W, Address = DSIM_BASE + 0x0018)

DSIM_MDRESOL	Bit	Description	Initial State
MainStandby	[31]	<p>Specifies standby for receiving DISPCON output in Command mode after setting all configuration.</p> <p>0 = Not ready 1 = Stand by</p> <p>Standby should be set after configuration (resolution, reqtype, pixelform, and so on) is set for command mode.</p> <p>In Video mode, if this bit value is 0, data is not transferred.</p>	0
Reserved	[30:28]	Reserved	-
MainVResol[11:0]	[27:16]	Specifies Vertical resolution (1 ~ 1024).	0x300
Reserved	[15:12]	Reserved	-
MainHResol[11:0]	[11:0]	Specifies Horizontal resolution (1 ~ 2047).	0x400

Main Display VPORCH Register (DSIM_MVPORCH, R/W, Address = DSIM_BASE + 0x001C)

DSIM_MVPORCH	Bit	Description	Initial State
CrndAllow	[31:28]	Specifies the number of horizontal lines, where command packet transmission is allowed after Stable VFP period.	0xF
Reserved	[27]	Reserved	-
StableVfp[10:0]	[26:16]	Specifies the number of horizontal lines, where command packet transmission is not allowed after end of active region. *Note: In Command mode, these bits are ignored.	0
Reserved	[15:11]	Reserved	-
MainVbp[10:0]	[10:0]	Specifies vertical back porch width for Video mode (line count). In Command mode, these bits are ignored.	0

* Transfers command packets after Stable VFP area. Display controller VFP lines should be set based on sum of these values: Stable VFP, command allowing area and command masked area. See the section for transferring general data in Video mode.

Main Display HPORCH Register (DSIM_MHPORCH, R/W, Address = DSIM_BASE + 0x0020)

DSIM_MHPORCH	Bit	Description	Initial State
MainHfp[15:0]	[31:16]	<p>Specifies the horizontal front porch width for Video mode. HFP is specified using blank packet.</p> <p>These bits specify the word counts for blank packet in HFP. In Command mode, these bits are ignored.</p>	0
MainHbp[15:0]	[15:0]	<p>Specifies the horizontal back porch width for Video mode. HBP is specified using blank packet.</p> <p>These bits specify the word counts for blank packet in HBP. In Command mode, these</p>	0

DSIM_MHPORCH	Bit	Description	Initial State
		bits are ignored.	

Main Display Sync Area Register (DSIM_MSYNC, R/W, Address = DSIM_BASE + 0x0024)

DSIM_MSYNC	Bit	Description	Initial State
MainVsa[9:0]	[31:22]	Specifies the vertical sync pulse width for Video mode (Line count). In command mode, these bits are ignored.	0
Reserved	[21:16]	Reserved	-
MainHsa[15:0]	[15:0]	Specifies the horizontal sync pulse width for Video mode. HSA is specified using blank packet. These bits specify word counts for blank packet in HSA. In command mode, these bits are ignored.	0

Sub Display Image Resolution Register (DSIM_SDRESOL, R/W, Address = DSIM_BASE + 0x0028)

DSIM_SDRESOL	Bit	Description	Initial State
SubStandby	[31]	Specifies standby for receiving DISPCON output in Command mode after setting all configuration. 0 = Not ready 1 = Standby Standby should be set after configuration (resolution, reqtype, pixelform, and so on) is set for command mode. In Video mode, this bit is ignored.	0
Reserved	[30:27]	Reserved	-
SubVResol[10:0]	[26:16]	Specifies the Vertical resolution (1 ~ 1024).	0x300
Reserved	[15:11]	Reserved	-
SubHResol[10:0]	[10:0]	Specifies the Horizontal resolution (1 ~ 1024).	0x400

Interrupt Source Register (DSIM_INTSRC, R/W, Address = DSIM_BASE + 0x002C)

This register identifies interrupt sources.

Internal block error, data transmit interrupt, inter-layer (D_PHY) error, etc.

The bits are set even if they are masked off by DSIM_INTMSK_REG.

Write '1' to clear the Interrupt.

DSIM_INTSRC	Bit	Description	Initial State
PllStable	[31]	Indicates that D-phy PLL is stable.	0
SwRstRelease	[30]	Releases the software reset.	0
SFRPLFifoEmpty	[29]	Specifies the SFR payload FIFO empty.	0
SFRPHFifoEmpty	[28]	Specifies the SFR Packet Header FIFO empty	0
SyncOverride	[27]	Indicates that other DSI command transfer have overridden sync timing.	0
Reserved	[26]	Reserved	-
BusTurnOver	[25]	Indicates when bus grant turns over from DSI slave to DSI master.	0
FrameDone	[24]	Indicates when MIPI DSIM transfers the whole image frame. Note: If Hsync is not received during two line times, internal timer is timed out and this bit is flagged.	0
Reserved	[23:22]	Reserved	-

DSIM_INTSRC	Bit	Description	Initial State
LpdrTout	[21]	Specifies the LP Rx timeout. See time out register (0x10).	0
TaTout	[20]	Turns around Acknowledge Timeout. See time out register (0x10).	0
Reserved	[19]	Reserved	-
RxDatDone	[18]	Completes receiving data.	0
RxTE	[17]	Receives TE Rx trigger.	0
RxAck	[16]	Receives ACK Rx trigger.	0
ErrRxECC	[15]	Specifies the ECC multi bit error in LPDR.	0
ErrRxCRC	[14]	Specifies the CRC error in LPDR.	0
ErrEsc3	[13]	Specifies the escape mode entry error lane 3. For more information, refer to standard D-PHY specification.	0
ErrEsc2	[12]	Specifies the escape mode entry error lane 2. For more information, refer to standard D-PHY specification.	0
ErrEsc1	[11]	Specifies the escape mode entry error lane 1. For more information, refer to standard D-PHY specification.	0
ErrEsc0	[10]	Specifies the escape mode entry error lane 0. For more information, refer to standard D-PHY specification.	0
ErrSync2	[9]	Specifies the LPDT sync error lane 3. For more information, refer to standard D-PHY specification.	0
ErrSync2	[8]	Specifies the LPDT Sync Error lane2. For more information, refer to standard D-PHY specification.	0
ErrSync1	[7]	Specifies the LPDT Sync Error lane1. For more information, refer to standard D-PHY specification.	0
ErrSync0	[6]	Specifies the LPDT Sync Error lane0. For more information, refer to standard D-PHY specification.	0
ErrControl2	[5]	Controls Error lane3. For more information, refer to standard D-PHY specification.	0
ErrControl2	[4]	Controls Error lane2. For more information, refer to standard D-PHY specification.	0
ErrControl1	[3]	Controls Error lane1. For more information, refer to standard D-PHY specification.	0
ErrControl0	[2]	Controls Error lane0. For more information, refer to standard D-PHY specification.	0
ErrContentLP0	[1]	Specifies the LP0 Contention Error (only lane0, because BTA occurs at lane0 only). For more information, refer to standard D-PHY specification.	0
ErrContentLP1	[0]	Specifies the LP1 Contention Error (only lane0, because BTA occurs at lane0 only). For more information, refer to standard D-PHY specification.	0

Interrupt Mask Register (DSIM_INTMSK, R/W, Address = DSIM_BASE + 0x0030)

This register masks interrupt sources.

DSIM_INTMSK	Bit	Description	Initial State
MskPllStable	[31]	Indicates that D-PHY PLL is stable.	1
MskSwRstRelease	[30]	Releases software reset.	0
MskSFRPLFifoEmpty	[29]	Empties SFR payload FIFO.	1
MskSFRPHFifoEmpty	[28]	Interrupt Mask for SFR packet header FIFO empty	1
MskSyncOverride	[27]	Indicates that other DSI command transfer have overridden sync timing.	1
Reserved	[26]	Reserved	-
MskBusTurnOver	[25]	Indicates when bus grant turns over from DSI slave to DSI master.	1

DSIM_INTMSK	Bit	Description	Initial State
MskFrameDone	[24]	Indicates when MIPI DSIM transfers whole image frame.	1
Reserved	[23:22]	Reserved	-
MskLpdrTout	[21]	Specifies LP Rx timeout. See time out register (0x10).	1
MskTaTout	[20]	Specifies turnaround acknowledge timeout. See time out register (0x10)	1
Reserved	[19]	Reserved	-
MskRxDatDone	[18]	Specifies completion of data receiving.	1
MskRxTE	[17]	Specifies receipt of TE Rx trigger.	1
MskRxAck	[16]	Specifies receipt of ACK Rx trigger.	1
MskRxECC	[15]	Specifies ECC multibit error in LPDR.	1
MskRxCRC	[14]	Specifies CRC error in LPDR.	1
MskEsc3	[13]	Specifies escape mode entry error in lane3. For more information, refer to standard D-PHY specification.	1
MskEsc2	[12]	Specifies escape mode entry error in lane2. For more information, refer to standard D-PHY specification.	1
MskEsc1	[11]	Specifies escape mode entry error in lane1. For more information, refer to standard D-PHY specification.	1
MskEsc0	[10]	Specifies escape mode entry error in lane0. For more information, refer to standard D-PHY specification.	1
MskSync3	[9]	Specifies LPDT sync error in lane3. For more information, refer to standard D-PHY specification.	1
MskSync2	[8]	Specifies LPDT sync error in lane2. For more information, refer to standard D-PHY specification.	1
MskSync1	[7]	Specifies LPDT sync error in lane1. For more information, refer to standard D-PHY specification.	1
MskSync0	[6]	Specifies LPDT sync error in lane0. For more information, refer to standard D-PHY specification.	1
MskControl3	[4]	Controls error in lane3. For more information, refer to standard D-PHY specification.	1
MskControl2	[4]	Controls error in lane2. For more information, refer to standard D-PHY specification.	1
MskControl1	[3]	Controls error in lane1. For more information, refer to standard D-PHY specification.	1
MskControl0	[2]	Controls error in lane0. For more information, refer to standard D-PHY specification.	1
MskContentLP0	[1]	Specifies LP0 contention error. For more information, refer to standard D-PHY specification.	1
MskContentLP1	[0]	Specifies LP1 contention error. For more information, refer to standard D-PHY specification.	1

Packet Header FIFO Register (DSIM_PKTHDR, W, Address = DSIM_BASE + 0x0034)

This register is the FIFO for packet header to send DSI packets.

DSIM_PKTHDR	Bit	Description	Initial State
Reserved	[31:24]	Reserved	-
PacketHeader	[23:0]	Writes the packet header of Tx packet. [7:0] = DI [15:8] = Dat0 (Word Count lower byte for long packet) [23:16] = Dat1 (Word Count upper byte for long packet)	0

Payload FIFO Register (DSIM_PAYLOAD, W, Address = DSIM_BASE + 0x0038)

This register specifies the FIFO for payload to send DSI packets.

DSIM_PAYLOAD	Bit	Description	Initial State
Payload	[31:0]	Writes the Payload of Tx packet.	0

Read FIFO Register (DSIM_RXFIFO, R, Address = DSIM_BASE + 0x003C)

This register is the gate of FIFO read

DSIM_RXFIFO	Bit	Description	Initial State
RxDat	[31:0]	In the Rx mode, you can read Rx data through this register. Note that the CRC in packet is not stored in Rx FIFO.	Unknown

FIFO Threshold Level Register (DSIM_FIFOTHLD, R/W, Address = DSIM_BASE + 0x0040)

Register	Address	Type	Description	Reset Value
DSIM_FIFOTHLD	Base + 0x0040	R/W	Threshold level register.	0x0000_01FF

DSIM_FIFOTHLD	Bit	ARM	Description	Initial State
	[31:9]	-	Reserved	0x0
WfullLevelSfr	[8:0]	R/W	Almost full level of SFR payload FIFO	0x1FF

FIFO Status & Control Register (DSIM_FIFOCTRL, R/W, Address = DSIM_BASE + 0x0044)

Register	Address	Type	Description	Reset Value
DSIM_FIFOCTRL	Base + 0x0044	R/W	FIFO status and control register.	0x0155_551F

DSIM_FIFOCTRL	Bit	ARM	Description	Initial State
	[31:26]		Reserved	0x0
FullRx	[25]	R	Rx FIFO full	0x0
EmptyRx	[24]	R	Rx FIFO empty	0x1
FullHSfr	[23]	R	SFR packet header FIFO full	0x0
EmptyHSfr	[22]	R	SFR packet header FIFO empty	0x1
FullLSfr	[21]	R	SFR payload FIFO full	0x0
EmptyLSfr	[20]	R	SFR payload FIFO empty	0x1
FullHI80	[19]	R	I80 packet header FIFO full	0x0
EmptyHI80	[18]	R	I80 packet header FIFO empty	0x1
FullLI80	[17]	R	I80 payload FIFO full	0x0
EmptyLI80	[16]	R	I80 payload FIFO empty	0x1
FullHSub	[15]	R	Sub display packet header FIFO full	0x0
EmptyHSub	[14]	R	Sub display packet header FIFO empty	0x1
FullLSub	[13]	R	Sub display payload FIFO full	0x0
EmptyLSub	[12]	R	Sub display payload FIFO empty	0x1
FullHMain	[11]	R	Main display packet header FIFO full	0x0

EmptyHMain	[10]	R	Main display packet header FIFO empty	0x1
FullLMain	[9]	R	Main display payload FIFO full	0x0
EmptyLMain	[8]	R	Main display payload FIFO empty	0x1
	[7:5]		Reserved	0x0
nInitRx	[4]	R/W	MD FIFO read point initialize	0x1
nInitSfr	[3]	R/W	SFR FIFO write point initialize	0x1
nInit80	[2]	R/W	I80 FIFO write point initialize	0x1
nInitSub	[1]	R/W	SD FIFO write point initialize	0x1
nInitMain	[0]	R/W	MD FIFO write point initialize	0x1

FIFO Memory AC Characteristic Register (DSIM_MEMACCHR, R/W, Address = DSIM_BASE + 0x0048)

Register	Address	Type	Description	Reset Value
DSIM_MEMACCHR	Base + 0x0048	R/W	FIFO memory AC characteristic register.	0x0000_4040

DSIM_MEMACCHR	Bits	ARM	Description	Initial State
	[31:16]		Reserved	0x00
PGEN_SD	[15]	R/W	Sub display FIFO memory power gating	0x0
RETN_SD	[14]	R/W	Sub display FIFO memory Retention	0x1
EMAB_SD	[13:11]	R/W	Sub display FIFO memory B port margin adjustment	0x0
EMAA_SD	[10:8]	R/W	Sub display FIFO memory A port margin adjustment	0x0
PGEN_MD	[7]	R/W	Main display FIFO memory power gating	0x0
RETN_MD	[6]	R/W	Main display FIFO memory Retention	0x1
EMAB_MD	[5:3]	R/W	Main display FIFO memory B port margin adjustment	0x0
EMAA_MD	[2:0]	R/W	Main display FIFO memory A port margin adjustment	0x0

*In current design, these memory port control was disabled. User can ignore the function of this register.

PLL Control Register (DSIM_PLLCTRL, R/W, Address = DSIM_BASE + 0x004C)

This register configures PLL control, D-PHY, clock range indication, and so on.

DSIM_PLLCTRL	Bit	Description	Initial State
Reserved	[31:28]	Should be 0.	-
BandCtrl	[27:24]	Each bandwidth control registers for Global Operation Timing. 0xF: 1Ghz 0xC: 750Mhz	
PllEn	[23]	Enables PLL.	0
Reserved	[22:20]	Should be 0.	-
PMS[19:1]	[19:1]	Specifies the PLL PMS value. 0x33E8: 1Ghz 0x43E8: 750Mhz	0
Reserved	[0]	Reserved	0

PLL Timer Register (DSIM_PLLTMR, R/W, Address = DSIM_BASE + 0x0050)

DSIM_PLLTMR	Bit	Description	Initial State
PITimer	[31:0]	Specifies the PLL Timer for stability of the generated clock (System clock cycle base). If the timer value goes to 0x00000000, the clock stable bit of status and interrupt register is set.	0xFFFFFFFF

PLL Control Register 1 (DSIM_PLLCTRL1, R/W, Address = DSIM_BASE + 0x0054)

This register configures D-PHY PLL control (M_PLLCTL[31:0])

DSIM_PLLCTRL1	Bit	Description	Initial State
M_PLLCTL0	[31:0]	It must be 0	0x0000_0000

PLL Control Register 2 (DSIM_PLLCTRL2, R/W, Address = DSIM_BASE + 0x0058)

This register configures D-PHY PLL control (M_PLLCTL[39:32])

DSIM_PLLCTRL2	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x000000
M_PLLCTL1	[7:0]	It must be 0	0x00

D-PHY Control Register (DSIM_PHYCTRL, R/W, Address = DSIM_BASE + 0x005C)

D-PHY Master & Slave Analog block characteristics control register (B_DPHYCTL).

DSIM_PHYCTRL	Bit	Description	Initial State
B_DPHYCTL	[31:0]	B_DPHYCTL[31:0] to D-PHY	0x0000_0000

D-PHY Control Register 1 (DSIM_PHYCTRL1, R/W, Address = DSIM_BASE + 0x0060)

D-PHY Master Analog block characteristics control register (M_DPHYCTL).

DSIM_PHYCTRL1	Bit	Description	Initial State
M_DPHYCTL	[31:0]	M_DPHYCTL[31:0] to D-PHY	0x0000_0000

D-PHY Timing Register (DSIM_PHYTIMING, R/W, Address = DSIM_BASE + 0x0064)

D-PHY Master global operating timing register.

DSIM_PHYTIMING	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0x0000
M_TLPXCTL	[15:8]	M_TLPXCTL[7:0] to D-PHY	0x00
M_THSEXITCTL	[7:0]	M_THSEXITCTL[7:0] to D-PHY	0x00

D-PHY Timing Register 1 (DSIM_PHYTIMING1, R/W, Address = DSIM_BASE + 0x0068)

D-PHY Master global operating timing register.

DSIM_PHYTIMING1	Bit	Description	Initial State
M_TCLKPRPCTL	[31:24]	M_TCLKPRPCTL[7:0] to D-PHY	0x00
M_TCLKZEROCTL	[23:16]	M_TCLKZEROCTL[7:0] to D-PHY	0x00
M_TCLKPOSTCTL	[15:8]	M_TCLKPOSTCTL[7:0] to D-PHY	0x00
M_TCLKTRAILCTL	[7:0]	M_TCLKTRAILCTL[7:0] to D-PHY	0x00

D-PHY Timing Register 2 (DSIM_PHYTIMING2, R/W, Address = DSIM_BASE + 0x006C)

D-PHY Master global operating timing register.

DSIM_PHYTIMING2	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
M_THSPRPRCTL	[23:16]	M_THSPRPRCTL[7:0] to D-PHY	0x00
M_THSZEROCTL	[15:8]	M_THSZEROCTL[7:0] to D-PHY	0x00
M_THSTRAILCTL	[7:0]	M_THSTRAILCTL[7:0] to D-PHY	0x00

Version Register (DSIM_VERSION, R, Address = DSIM_BASE + 0x0070)

DSIM_VERSION	Bit	Description	Initial State
Version	[31:0]	Specifies the DSIM version information	0x8000_0001

Stereo Scope 3D Register (DSIM_S3D_CTL, R/W, Address = DSIM_BASE + 0x0080)

DSIM_S3D_CTL	Bit	Description	Initial State
Reserved	[31:12]	Reserved	0x00000
3DPRESENT	[11]	Stereo Scope 3D control payload is present	0
Reserved	[10:6]	Reserved	0x0
3DL/R	[5]	Left / Right Order '0' = Data sent left eye first, right eye next. '1' = Data sent right eye first, left eye next.	0
3DVSYNC	[4]	Second VSYNC Enabled between Left and Right Images '0' = No sync pulses between left and right data. '1' = Sync pulse (HSYNC, VSYNC, blanking) between left and right data.	0
3DFMT	[1:0]	3D Image Format '00' = Line (alternating lines of left and right data). '01' = Frame (alternating frames of left and right data). '10' = Pixel (alternating pixels of left and right data). '11' = Reserved	0x0
3DMODE	[1:0]	3D Mode On / Off, Display Orientation '00' = 3D Mode Off (2D Mode On). '01' = 3D Mode On, Portrait Orientation. '10' = 3D Mode On, Landscape Orientation. '11' = Reserved.	0

Data ID(0x01, VSYNC Start), Fixed							
Data 0	reserved	reserved	reserved	reserved	3DPRESENT	reserved	reserved
Data 1	0	0	3DL/R	3DVSYNC	3DFMT[1:0]	3DMODE[1:0]	

Proprietary 3D Register (DSIM_P3D_CTL, R/W, Address = DSIM_BASE + 0x0084)

DSIM_P3D_CTL	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
Data_ID	[23:16]	Data type for processor-sourced packet data type 0x23 is generic short write, 2 parameters	0x23
P3D_ID	[15:8]	Proprietary 3D ID	0x4D
Reserved	[7:6]	Reserved	0x0
P3D_Mode	[5:4]	Proprietary 3D mode '00' = Sub-pixel mode '01' = Side-by-Side mode '10','11' = Reserved	0x0
Reserved	[3:2]	Reserved	0x0
P3D_EN	[1]	Proprietary 3D enable / disable If this bit is 0, the peripheral may ignore the Proprietary 3D Register.	0
P3D_On_Off	[0]	Proprietary 3D On / Off If 3D On from FIMD, It can be change to '1'. This bit is read only.	0

Data ID	Data ID(default : 0x23, Generic short write, 2parameters)					
Data 0	SEC_3D_CTRL(default : 0x4D)					
Data 1	reserved	reserved	proprietary 3d mode	reserved	reserved	Enable / Disable

MIC Register (DSIM_MIC_CTL, R/W, Address = DSIM_BASE + 0x0088)

DSIM_P3D_CTL	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
Data_ID	[23:16]	Data type for processor-sourced packet data type 0x23 is generic short write, 2 parameters	0x23
MIC_ID	[15:8]	MIC ID	0x4D
Reserved	[7:2]	Reserved	0x00
MIC_EN	[1]	MIC enable / disable If this bit is 0, the peripheral may ignore the MIC Register.	0
MIC_On_Off	[0]	MIC On / Off If MIC On from FIMD, It can be change to '1'. This bit is read only.	0

Data ID	Data ID(default : 0x23, Generic short write, 2parameters)						
Data 0	SEC_MIC_CTRL(default : 0x4F)						
Data 1	reserved	reserved	reserved	reserved	reserved	reserved	Enable / Disable

Proprietary On MIC Off Horizontal Register (DSIM_P3D_ON_MIC_OFF_HORIZONTAL, R/W, Address = DSIM_BASE + 0x008C)

DSIM_P3D_ON_MIC_OFF_HORIZONTAL	Bit	Description	Initial State
Reserved	[31:12]	Reserved	0x00000
H_Size	[11:0]	Horizontal size when Proprietary 3D On, MIC Off	0x400

Proprietary Off MIC On Horizontal Register (DSIM_P3D_OFF_MIC_ON_HORIZONTAL, R/W, Address = DSIM_BASE + 0x0090)

DSIM_P3D_OFF_MIC_ON_HORIZONTAL	Bit	Description	Initial State
Reserved	[31:12]	Reserved	0x00000
H_Size	[11:0]	Horizontal size when Proprietary 3D Off, MIC On	0x400

Proprietary On MIC On Horizontal Register (DSIM_P3D_ON_MIC_ON_HORIZONTAL, R/W, Address = DSIM_BASE + 0x0094)

DSIM_P3D_ON_MIC_ON_HORIZONTAL	Bit	Description	Initial State
Reserved	[31:12]	Reserved	0x00000
H_Size	[11:0]	Horizontal size when Proprietary 3D On, MIC On	0x400

Proprietary On MIC Off HFP Register (DSIM_P3D_ON_MIC_OFF_HFP, R/W, Address = DSIM_BASE + 0x0098)

DSIM_P3D_ON_MIC_OFF_HFP	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0x0000
HFP_Size	[15:0]	HFP size when Proprietary 3D On, MIC Off for Video mode. HFP is specified using blank packet. In Command mode, these bits are ignored.	0x0000

Proprietary Off MIC On HFP Register (DSIM_P3D_OFF_MIC_ON_HFP, R/W, Address = DSIM_BASE + 0x009C)

DSIM_P3D_OFF_MIC_ON_HFP	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0x0000
HFP_Size	[15:0]	HFP size when Proprietary 3D Off, MIC On for Video mode. HFP is specified using blank packet.	0x0000

DSIM_P3D_OFF_MIC_ON_HFP	Bit	Description	Initial State
		In Command mode, these bits are ignored.	

Proprietary On MIC On HFP Register (DSIM_P3D_ON_MIC_ON_HFP, R/W, Address = DSIM_BASE + 0x00A0)

DSIM_P3D_ON_MIC_ON_HFP	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0x0000
HFP_Size	[15:0]	HFP size when Proprietary 3D On, MIC On for Video mode. HFP is specified using blank packet. In Command mode, these bits are ignored.	0x0000

38.3 CSIS

38.3.1 Interfaces and Protocol

38.3.1.1 D-PHY layer FSM

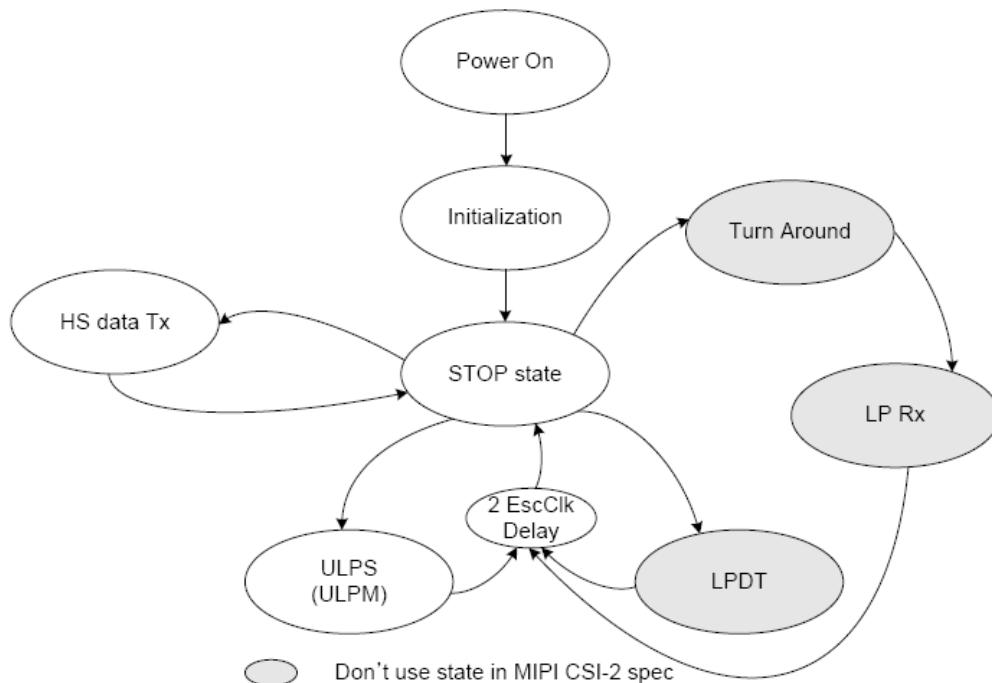


Figure 38-14. D-PHY Finite State Machine (Tx D-phy of Samsung)

MIPI CSIS V3.0 system supports HSDT(High Speed Data transfer) and ULPS(ULPM – Ultra-Low Power State or Mode) only. There is no trigger function, LPDT and BTA.

38.3.1.2 PPI interface timing & protocol

MIPI CSIS V3.0 supports HSDT and ULPM.

High Speed Data Transfer

In Figure 38-15, the upper 5 signals are related with clock lane and the lower 6 signals are related with data lane. Dp and Dn of upper signals are D-phy channel of clock lane. RX_DDRCLKDIV2 is in PPI. BYTE_CLK is generated clock that is generated in link layer(MIPI CSIS V3.0) and divided by 2 from RX_DDRCLKDIV2. Dp and Dn of lower signals are D-phy channel of Data lane. Another signals of lower are PPI. STOPstate(and STOPstateClk) indicates that differential channel state of D-phy is LP11(the voltage level of Dp and Dn is 1.2V)

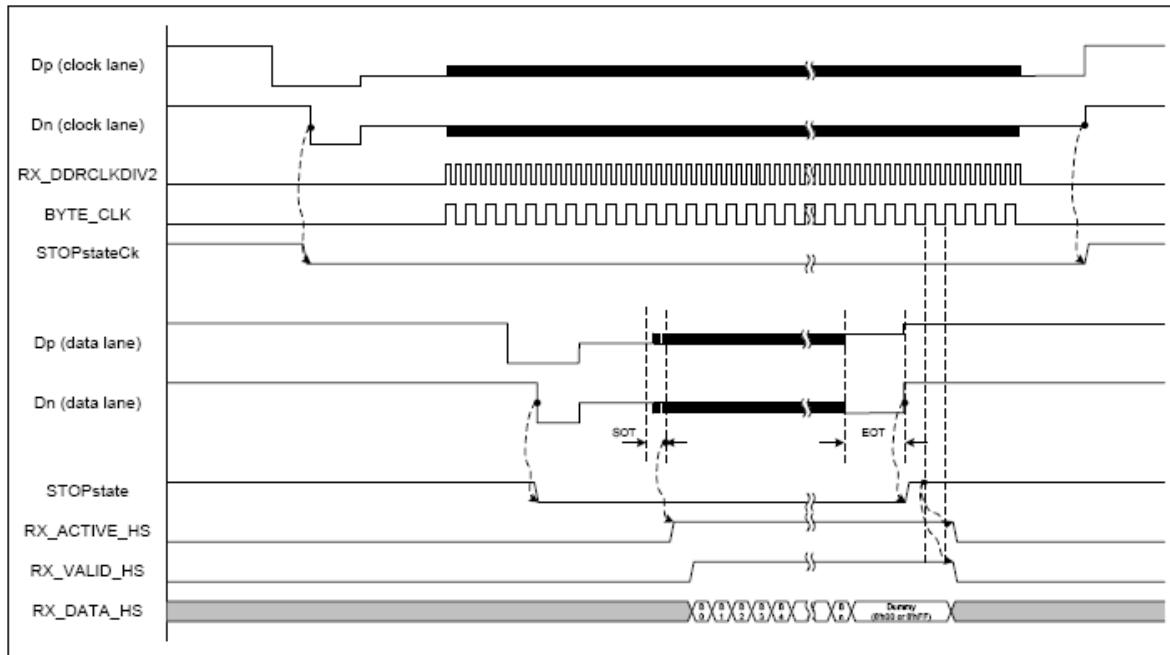


Figure 38-15. Timing Diagram of High Speed Data Transfer

Ultra-Low Power Mode

In Figure 38-16, UlpsActiveNot* and STOPstate* is in PPI. ULPS command in Data lane is only D-phy command that is generated Tx D-phy. Rx D-phy decodes this ULPS command and generates Ulps

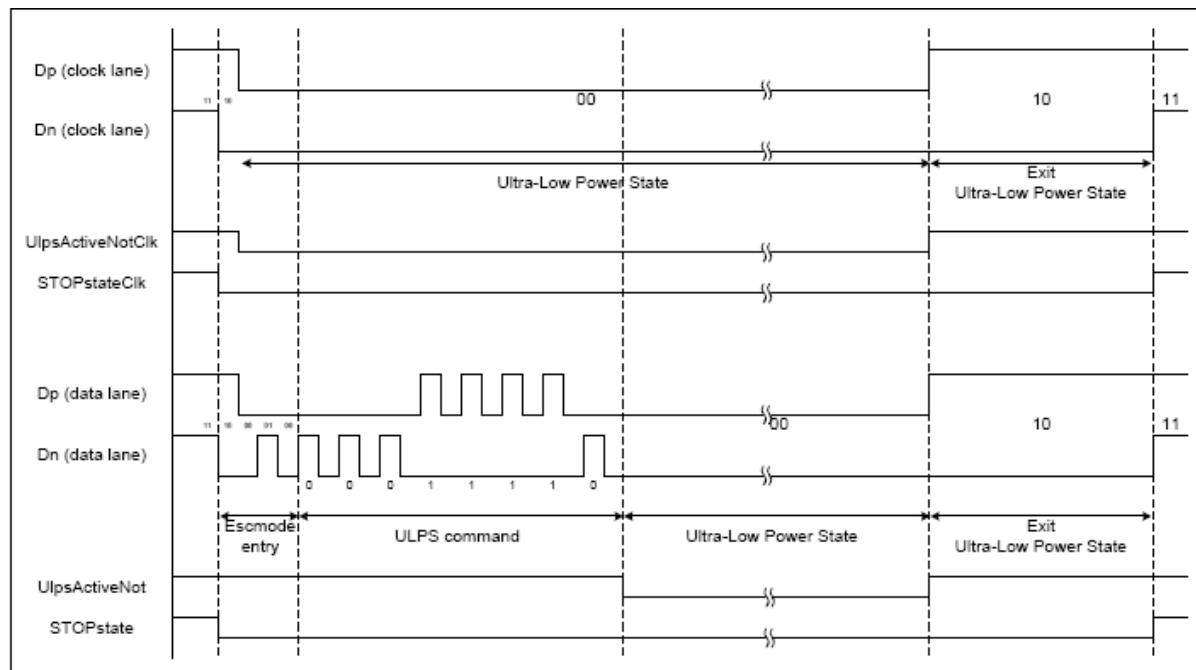


Figure 38-16. Timing Diagram of Ultra Low Power Mode

ISP(CAM I/F) interface

MIPI CSIS V3.0 output signals are PIX_CLK, VVALID, HVALID, and DATA. PIX_CLK is output pixel clock what is

generated from HCLK, BYTE_CLK or EXTCLK. VVALID is vertical sync signal. HVALID is horizontal sync signal. DATA is image data bus. DATA bus width is dependent on Image format. Maximum bus width is 24bits because of RGB888. Figure 38-17 describes the output protocol of MIPI CSIS. All signal is synchronized with the rising edge of PIX_CLK.

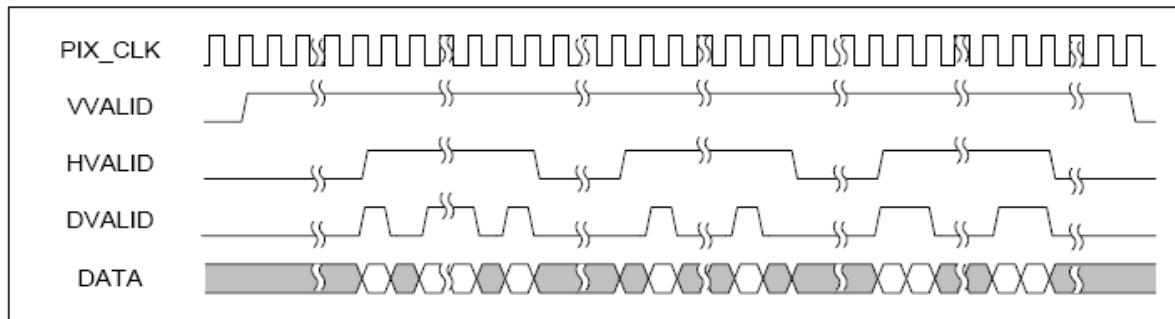


Figure 38-17. Output protocol of ISP Wrapper of MIPI CSIS V3.0

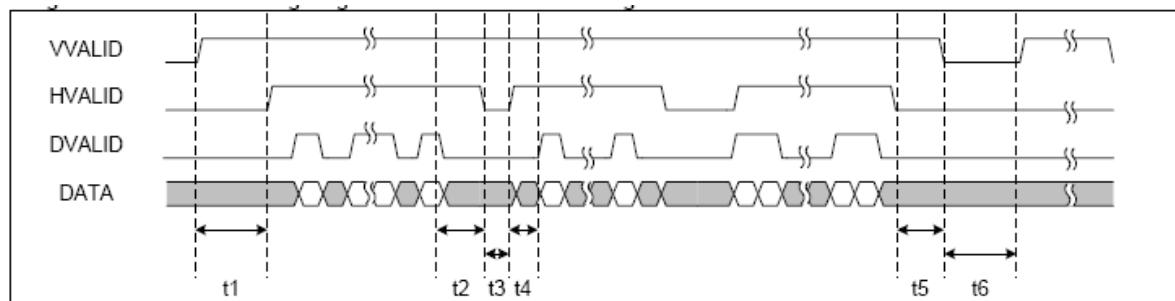


Figure 38-18. Timing diagram of Output Protocol of ISP warpper

	Description	Minimum cycle of pixel clock	Maximum cycle of pixel clock
t1	Interval between rising of VVALID and first rising of HVALID	Vsync_SIntv + 1 (1 ~ 64)	-
t2	Interval between last falling of DVALID and falling of HVALID	Hsync_LIntv + 2 (2 ~ 66)	-
t3	Interval between falling of HVALID and rising of next HVALID	1	-
t4	Interval between rising of HVALID and first rising of DVALID	0	-
t5	Interval between last falling of HVALID and falling of VVALID	Vsync_EIntv (0 ~ 4095)	-
t6	Interval between falling of VVALID and rising of next VVALID	1	-

Description of Output Protocol

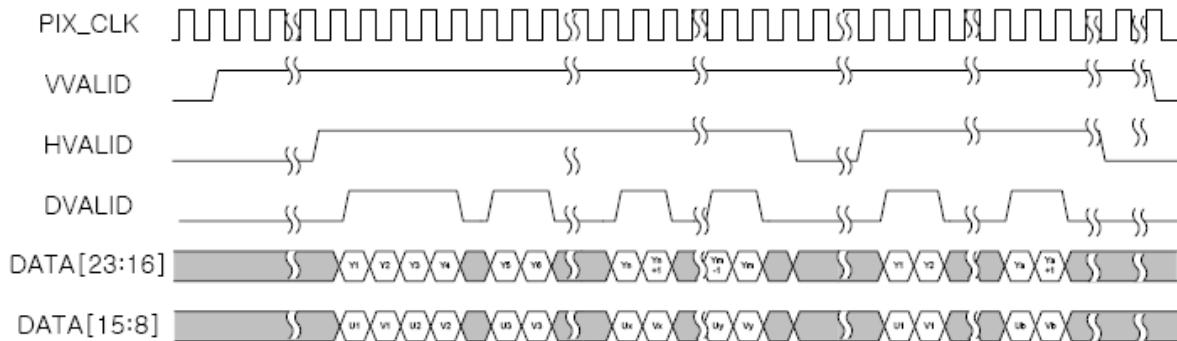


Figure 38-19. Waveform of ISP interface (CAM I/F)

38.3.2 Configuration

Image resolution

MIPI CSI Slave block needs the configuration of Image resolution to measure Hsync pulse length exactly and detect frame end.

Vertical resolution register has 16bits (16" h0001 ~ 16" hFFFF)

Horizontal resolution register has 16bits (16" h0001 ~ 16" hFFFF).

Image data format

NXP4330D/Q supports YUV422 8bit format only.

Data type	Description
0x18	YUV420 8-bit
0x19	YUV420 10-bit
0x1A	Legacy YUV420 8-bit
0x1B	Reserved
0x1C	YUV420 8-bit CSPS (Chroma Shifted Pixel Sampling)
0x1D	YUV420 10-bit CSPS (Chroma Shifted Pixel Sampling)
0x1E	YUV422 8-bit
0x1F	YUV422 10-bit

38.3.3 Interrupt

MIPI CSIS V3.0 has many interrupts for checking status, indicating error case and receiving generic data.

Odd_Before/Odd_After/Even_Before/Even_After

These interrupts are related with generic and embedded data.

Odd_Before and Odd_After interrupts are generated in odd frame. Even_Before and Even_After interrupts are generated in even frame. Odd_Before and Even_Before interrupts are generated when Generic Short packet or Embedded 8-bit based packet is received before image data (Vertical Back porch area)

38.3.4 Clock specification

MIPI CSI may have 3 clock sources: RX_BYTE_CLK_HS0, I_PCLK, and I_WRAP_CLK.

I_PCLK	PCLK is system clock generated by general processor PLL(APB clock)
RX_BYTE_CLK_HS0	This signal comes from data lane 0 of D-phy.
I_WRAP_CLK	I_WRAP_CLK is generated by clock generator. Refer to CSI clock in clock controller.

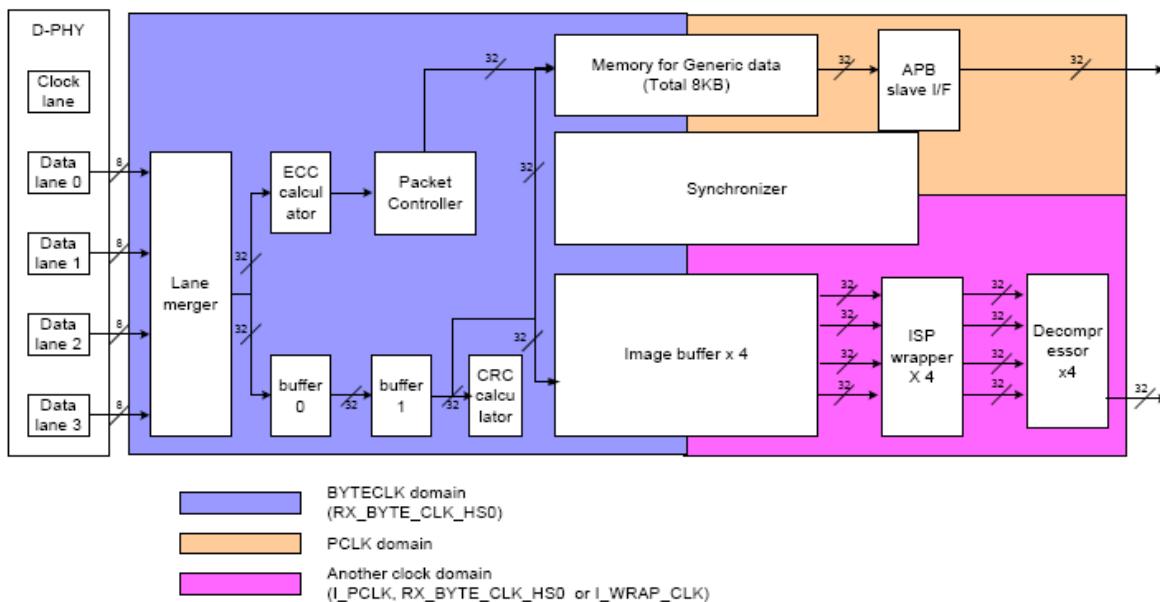


Figure 38-20. Block Diagram of Clock Domain

All clock domains are asynchronous each other. Pixel clock for transmitting image data to ISP is same with HCLK or Wrapper clock.

The relationship between input and output bandwidth is that the output bandwidth should be faster than input bandwidth. There is an equation of previous relationship. :

$$(\text{freq. of RX_BYTE_CLK_HS}) * (\text{number of data lane}) * 8\text{bits} \leq (\text{freq. of Pixel clock}) * (\text{bitwidth of image format})$$

38.3.5 CSIS Register Summary

BASE address : 0xC00D0000

Register map

Register name	Offset	Type	Description	Reset Value
CSIS_CTRL	0xC00D0000	R/W	Control register	0x0010_0000
CSIS_DPHYCTRL	0xC00D0004	R/W	DPHY Analog Control register	0x0000_0000
CSIS_CONFIG_CH0	0xC00D0008	R/W	Configuration register of CH0	0x0000_00FC
CSIS_DPHYSTS	0xC00D000C	RO	DPHY Status register	0x0000_00F1
CSIS_INTMSK	0xC00D0010	R/W	Interrupt mask register	0x0000_0000
CSIS_INTSRC	0xC00D0014	R/W	Interrupt source register Control	0x0000_0000
CSIS_CTRL2	0xC00D0018	R/W	Control register about ch1~3	0x00E0_0000

Register name	Offset	Type	Description	Reset Value
CSIS_VERSION	0xC00D001C	RO	CSIS version register	0x8000_0000
CSIS_DPHYCTRL_0	0xC00D0020	R/W	DPHY Analog Control register0	0x0000_0000
CSIS_DPHYCTRL_1	0xC00D0024	R/W	DPHY Analog Control register1	0x0000_0000
RESERVED	0xC00D0028	R/W		0x0000_0000
CSIS_RESOL_CH0	0xC00D002C	R/W	Image Resolution register of CH0	0x8000_8000
RESERVED	0xC00D0030	R/W		0x0000_0000
RESERVED	0xC00D0034	R/W		0x8000_8000
SDW_CONFIG_CH0	0xC00D0038	R/W	Shadow register of CH0 Configuration	0x0000_00FC
SDW_RESOL_CH0	0xC00D003C	RO	Shadow register of CH0 Resolution	0x8000_8000
CSIS_CONFIG_CH1	0xC00D0040	R/W	Configuration register of CH1	0x0000_00FC
CSIS_RESOL_CH1	0xC00D0044	R/W	Image Resolution register of CH1	0x8000_8000
SDW_CONFIG_CH1	0xC00D0048	R/W	Shadow register of CH1 Configuration	0x0000_00FC
SDW_RESOL_CH1	0xC00D004C	RO	Shadow register of CH1 Resolution	0x8000_8000
CSIS_CONFIG_CH2	0xC00D0050	R/W	Configuration register of CH2	0x0000_00FC
CSIS_RESOL_CH2	0xC00D0054	R/W	Image Resolution register of CH2	0x8000_8000
SDW_CONFIG_CH2	0xC00D0058	R/W	Shadow register of CH2 Configuration	0x0000_00FC
SDW_RESOL_CH2	0xC00D005C	RO	Shadow register of CH2 Resolution	0x8000_8000
CSIS_CONFIG_CH3	0xC00D0060	R/W	Configuration register of CH3	0x0000_00FC
CSIS_RESOL_CH3	0xC00D0064	R/W	Image Resolution register of CH3	0x8000_8000
SDW_CONFIG_CH3	0xC00D0068	R/W	Shadow register of CH3 Configuration	0x0000_00FC
SDW_RESOL_CH3	0xC00D006C	RO	Shadow register of CH3 Resolution	0x8000_8000
DSIM	0xC00D0100 ~ 0xC00D01FF			
CSIS_NONIMG_ODD	0xC00D2000 ~ 0xC00D2FFF	R/W	Memory area for storing non-image data. Odd frame	0XXXX_XXXX
CSIS_NONIMG_EVEN	0xC00D3000 ~ 0xC00D3FFF	R/W	Memory area for storing non-image data. Even frame	0XXXX_XXXX

Register descriptions

Bit	R/W	Symbol	Description	Reset Value
Control Register(CSIS_CTRL)				
Address : C00D_0000h : WORD				
[31]	R/W	S_DPDN_SWAP_CLK	Swapping Dp and Dn channel of clock lane. 0 : default 1 : swapped	0
[30]	R/W	S_DPDN_SWAP_DAT	Swapping Dp and Dn channel of data lanes. 0 : default 1 : swapped	0
[29:28]		RESERVED	read as zero, do not modify	0

Bit	R/W	Symbol	Description	Reset Value
[27:26]	R/W	DECOMP_FORM	Decompress format 00 : 10 bit compressed format 01 : reserved for 12bit compressed format Do not modify 1x : reserved This register field related with Data format field in CSIS_CONFIG register	0
[25]	R/W	DECOMP_PREDICT	Decompress prediction mode of CH0 0 : simple prediction 1 : advanced prediction	0
[24]	R/W	DECOMP_EN	Decompress enable 0 : disable (default) 1 : enable When default value, input data of decompressor is bypassed with all protocol signals (Vvalid, Hvalid, Dvalid and Bvalid)	0
[23:22]	R/W	INTERLEAVE_MODE	Select Interleave mode, VC(Virtual channel) and DT(Data type) 3: VC and DT 2: VC only 1: DT only 0: CH0 only, no data interleave	0
[21]	R/W	DOUBLE_CMPNT	Double component per clock cycle in YUV422 formats, CH0 0 : single component per clock cycle (half pixel per clock cycle) 1 : double component per clock cycle (a pixel per clock cycle)	0
[20]	R/W	PARALLEL	Output bus width of CH0 is 32 bits. 0 : Normal output 1 : 32bit data alignment When this bit is set, the outer bus width of MIPI CSIS V3.0 is 32.	1
[19:16]	R/W	UPDATE_SHADOW	Strobe of updating shadow registers 0 : default 1 : update the shadow registers. After configuration, User has to set this bit for updating shadow registers. This bit is cleared automatically after updating shadow registers. Bit [19] CH3 to Bit[16]CH0	0
[15:12]	R/W	RGB_SWAP	Swapping RGB sequence 0: MSB is R and LSB is B. 1 : MSB is B and LSB is R. (swapped) Bit [15] CH3 to Bit[12]CH0	0
[11:8]	R/W	WCLK_SRC	Wrapper clock source 0: PCLK 1: L_WRAP_CLK This bit determines source of Pixel clock which is clock of transferring image data to ISP or CAM I/F. When data format is "User defined packet", this bit is ignored. Bit [11] CH3 to Bit[8]CH0	0
[7:5]		RESERVED	read as zero, do not modify	0
[4]	WO	SW_RST	Software reset	0

Bit	R/W	Symbol	Description	Reset Value
			0 : ready 1 : reset All writable registers in CSI2 go back to initial state. After this bit is active for 3 cycles, this bit will be de-asserted automatically * Notice : Almost MIPI CSI2 block uses "ByteClk" from D-phy. This "ByteClk" is not continuous clock. User has to assert software reset when Camera module is turned off.	
[3:2]	R/W	NUMOFDATALENE	Number of data lane 00 : 1 data lane 01 : 2 data lane 10 : 3 data lane 11 : 4 data lane	0
[1]		RESERVED	read as zero, do not modify	0
[0]	R/W	CSI_EN	MIPI CSI2 system enable 0 : disable 1 : enable	0
Configuration register for channel 0 (CSIS_CONFIG_CH0)				
Address : C00D_0008h : WORD				
[31:26]	R/W	HSYNC_LINTV_CH0	Interval between Hsync falling and Hsync rising (Line interval) 6'h00 ~ 6'h3F cycle of Pixel clock	0
[25:20]	R/W	VSYNC_SINTV_CH0	interval between Vsync rising and first Hsync rising. 6'h00 ~ 6'h3F cycle of Pixel clock	0
[19:8]	R/W	VSYNC_EINTV_CH0	interval between last Hsync falling and Vsync falling. 12'h000 ~ 12'hFFF cycle of Pixel clock	0
[7:2]	R/W	DATAFORMAT_CH0	Image Data Format YUV420 (8bit) : 0x18 YUV420 (10bit) : 0x19 YUV420 (8bit legacy) : 0x1A YUV420 (8bit CSPS) : 0x1C YUV420 (10bit CSPS) : 0x1D YUV422 (8bit) : 0x1E YUV422 (10bit) : 0x1F RGB565 : 0x22 RGB666 : 0x23 RGB888 : 0x24 RAW6 : 0x28 RAW7 : 0x29 RAW8 : 0x2A RAW10 : 0x2B RAW12 : 0x2C RAW14 : 0x2D User defined 1 : 0x30 User defined 2 : 0x31 User defined 3 : 0x32 User defined 4 : 0x33	0x3F
[1:0]	R/W	VIRTUAL_CHANNEL_CH0	Set Virtual channel for data interleave. 00: VC=0 01: VC=1	0

Bit	R/W	Symbol	Description	Reset Value
			10: VC=2 11: VC=3	
D-PHY status register (CSIS_DPHYSTS)				
Address : C00D_000Ch : WORD				
[31:12]		RESERVED		0
[11:8]	R	ULPSDAT	Data lane [3:0] is in ULPS [7] : data lane 3 [6] : data lane 2 [5] : data lane 1 [4] : data lane 0 0 : not ULPS 1 : ULPS	0
[7:4]	R	STOPSTATEDAT	Data lane [3:0] is in Stop State [7] : data lane 3 [6] : data lane 2 [5] : data lane 1 [4] : data lane 0 0 : not Stop state 1 : Stop state	0x1F
[3:2]		RESERVED		0
[1]	R	ULPSCLK	Clock lane is in ULPS 0 : not ULPS 1 : ULPS	0
[0]	R	STOPSTATECLK	Clock lane is in Stop State 0 : not Stop state 1 : Stop state	1
Interrupt mask register(CSIS_INTMSK)				
Address : C00D_0010h : WORD				
[31]	R/W	MSK_EVENBEFORE	Non Image data are received at Even frame and Before Image 0 : disable (masked) 1 : enable (unmasked)	0
[30]	R/W	MSK_EVENAFTER	Non Image data are received at Even frame and After Image 0 : disable (masked) 1 : enable (unmasked)	0
[29]	R/W	MSK_ODDBEFORE	Non Image data are received at Odd frame and Before Image 0 : disable (masked) 1 : enable (unmasked)	0
[28]	R/W	MSK_ODDAFTER	Non Image data are received at Odd frame and After Image 0 : disable (masked) 1 : enable (unmasked)	0
[27:24]	R/W	MSK_FRAMESTART	FS packet is received, [CH3,CH2,CH1,CH0] 0 : disable (masked) 1 : enable (unmasked)	0
[23:20]	R/W	MSK_FRAMEEND	FE packet is received, [CH3,CH2,CH1,CH0] 0 : disable (masked)	0

Bit	R/W	Symbol	Description	Reset Value
			1 : enable (unmasked)	
[19:17]		RESERVED	read as zero, do not modify	0
[16]	R/W	MSK_ERR_SOT_HS	Start of transmission error 0 : disable (masked) 1 : enable (unmasked)	0
[15:12]	R/W	MSK_ERR_LOST_FS	Lost of Frame Start packet, [CH3,CH2,CH1,CH0] 0 : disable (masked) 1 : enable (unmasked)	0
[11:8]	R/W	MSK_ERR_LOST_FE	Lost of Frame End packet, [CH3,CH2,CH1,CH0] 0 : disable (masked) 1 : enable (unmasked)	0
[7:4]	R/W	MSK_ERR_OVER	Image FIFO overflow interrupt, [CH3,CH2,CH1,CH0] 0 : disable (masked) 1 : enable (unmasked)	0
[3]		RESERVED	read as zero, do not modify	0
[2]	R/W	MSK_ERR_ECC	ECC error 0 : disable (masked) 1 : enable (unmasked)	0
[1]	R/W	MSK_ERR_CRC	CRC error 0 : disable (masked) 1 : enable (unmasked)	0
[0]	R/W	MSK_ERR_ID	Unknown ID error 0 : disable (masked) 1 : enable (unmasked)	0

Interrupt source register(CSIS_INTSRC) - Write 1 clears status bit and write 0 has no effect.

Address : C00D_0014h : WORD

[31]	R/W	EVENBEFORE	Non Image data are received at Even frame and Before Image	0
[30]	R/W	EVENAFTER	Non Image data are received at Even frame and After Image	0
[29]	R/W	ODDBEFORE	Non Image data are received at Odd frame and Before Image	0
[28]	R/W	ODDAFTER	Non Image data are received at Odd frame and After Image	0
[27:24]	R/W	FRAMESTART	FS packet is received, [CH3,CH2,CH1,CH0]	0
[23:20]	R/W	FRAMEEND	FE packet is received, [CH3,CH2,CH1,CH0]	0
[19:16]	R/W	ERR_SOT_HS	Start of transmission error, [CH3,CH2,CH1,CH0]	0
[15:12]	R/W	ERR_LOST_FS	Indication of lost of Frame Start packet, [CH3,CH2,CH1,CH0]	0
[11:8]	R/W	ERR_LOST_FE	Indication of lost of Frame End packet, [CH3,CH2,CH1,CH0]	0
[7:4]	R/W	ERR_OVER	Overflow is caused in image FIFO. [CH3,CH2,CH1,CH0] Outer bandwidth has to be faster than imputer bandwidth. But image FIFO can be overflow because of user.s fault. There are 2 ways for preventing overflow. 1. Tune output pixel clock faster than current. 2. Tune input byte clock slow than current. First case : WCLK_Src in CSIS_CTRL register shoud be set 1, and then assign faster clock Second case : user can set register in camera module	0

Bit	R/W	Symbol	Description	Reset Value
			through I2C channel. When this interrupt is generated, 1. Turn camera off 2. Assert software reset, if you didn't assert software reset, MIPI CSIS could not receive any more data. 3. Tune the clock frequency and re-configure all related registers. MIPI CSIS module is ready for operating.	
[3]		RESERVED	read as zero, do not modify	0
[2]	R/W	ERR_ECC	ECC error	0
[1]	R/W	ERR_CRC	CRC error	0
[0]	R/W	ERR_ID	Unknown ID error	0
Control register 2(CSIS_CTRL2)				
Address : C00D_0018h : WORD				
[31:29]	R/W	DECOMP_PREDICT	Decompress prediction mode of CH3 to CH1 0 : simple prediction 1 : advanced prediction Bit [31] : CH3, Bit [30] : CH2, Bit [29] : CH1	0
[28]		RESERVED	read as zero, do not modify	0
[27:25]	R/W	DOUBLE_CMPNT	Double component per clock cycle in YUV422 formats 0 : single component per clock cycle (half pixel per clock cycle) 1 : double component per clock cycle (a pixel per clock cycle) Bit [27] : CH3, Bit [26] : CH2, Bit [25] : CH1	0
[24]		RESERVED	read as zero, do not modify	0
[23:21]	R/W	PARALLEL	Output bus width of CH0 is 32 bits. 0 : Normal output 1 : 32bit data alignment When this bit is set, the outer buswidth of MIPI CSIS V3.0 is 32. Bit [23] : CH3, Bit [22] : CH2, Bit [21] : CH1	0x7
[20:0]		RESERVED	read as zero, do not modify	0
CSIS version register(CSIS_VERSION)				
Address : C00D_001Ch : WORD				
[31:0]	R	CSIS_VERSION	CSIS version information	0x8000_0000
D-PHY Analog Control register_0(B_DphyCtrl)				
Address : C00D_0020h : WORD				
[31:0]	R/W	B_DPHYCTRL	D-PHY Slave analog block characteristics control register. Do not modify this.	0
D-PHY Analog Control register_1 (S_DphyCtrl)				
Address : C00D_0024h : WORD				
[31:0]	R/W	S_DPHYCTRL	D-PHY Slave analog block characteristics control register. It must be 0.	0
Resolution of CH0 register (CSIS_RESOL_CH0)				
Address : C00D_002Ch : WORD				

Bit	R/W	Symbol	Description	Reset Value
[31:16]	R/W	HRESOL_CH0	Horizontal Image resolution Input boundary of each image format YUV420 (8bit) : 0x0001 ~ 0xFFFF YUV420 (10bit) : 4n (n is 1, 2, 3,...) YUV420 (8bit legacy) : 0x0001 ~ 0xFFFF YUV420 (8bit CSPS) : 0x0001 ~ 0xFFFF YUV420 (10bit CSPS) : 4n (n is 1, 2, 3,...) YUV422 (8bit) : 0x0001 ~ 0xFFFF YUV422 (10bit) : 4n (n is 1, 2, 3,...) RGB565 : 0x0001 ~ 0xFFFF RGB666 : 4n (n is 1, 2, 3,...) RGB888 : 0x0001 ~ 0xFFFF RAW8 : 0x0001 ~ 0xFFFF RAW10 : 4n (n is 1, 2, 3,...) RAW12 : 2n (n is 1, 2, 3,...) RAW14 : 4n (n is 1, 2, 3,...) System LSI Division, Semiconductor Business 26 Confidential Property of Samsung Electronics Co., Ltd. Internal User Only	0x8000
[15:0]	R/W	VRESOL_CH0	Vertical Image resolution Input boundary : 0x0001 ~ 0xFFFF	0x8000
Shadow Configuration of CH0 register (SDW_CONFIG_CH0)				
Address : C00D_0038h : WORD				
[31:26]	R	SDW_HSYNC_LINTV_CH0	Current interval between Hsync falling and Hsync rising (Line interval)	0
[25:20]	R	SDW_VSYNC_SINTV_CH0	Current interval between Vsync rising and first Hsync rising.	0
[19:8]	R	SDW_VSYNC_EINTV_CH0	Current interval between last Hsync falling and Vsync falling.	0
[7:2]	R	SDW_DATAFORMAT_CH0	Current image Data Format	0
[1:0]	R	SDW_VIRTUAL_CHANNEL_CH0	Set Virtual channel for data interleave	0
Shadow Resolution of CH0 register (SDW_RESOL_CH0)				
Address : C00D_003Ch : WORD				
[31:16]	R	SDW_HRESOL_CH0	Current Horizontal Image resolution	0x8000
[15:0]	R	SDW_VRESOL_CH0	Current Vertical Image resolution	0x8000
Channel 1 registers (CSIS_CONFIG_CH1, CSIS_RESOL_CH1, SDW_CONFIG_CH1, SDW_RESOL_CH1)				
Address : C00D_0040h, C00D_0044h, C00D_0048h, C00D_004Ch : WORD				
Each register has same format with channel 0 register.				

38.4 D-PHY

38.4.1 Architecture

38.4.1.1 PLL and Clock Lane Connection

The following figure illustrates the PLL and Clock Lane connection.

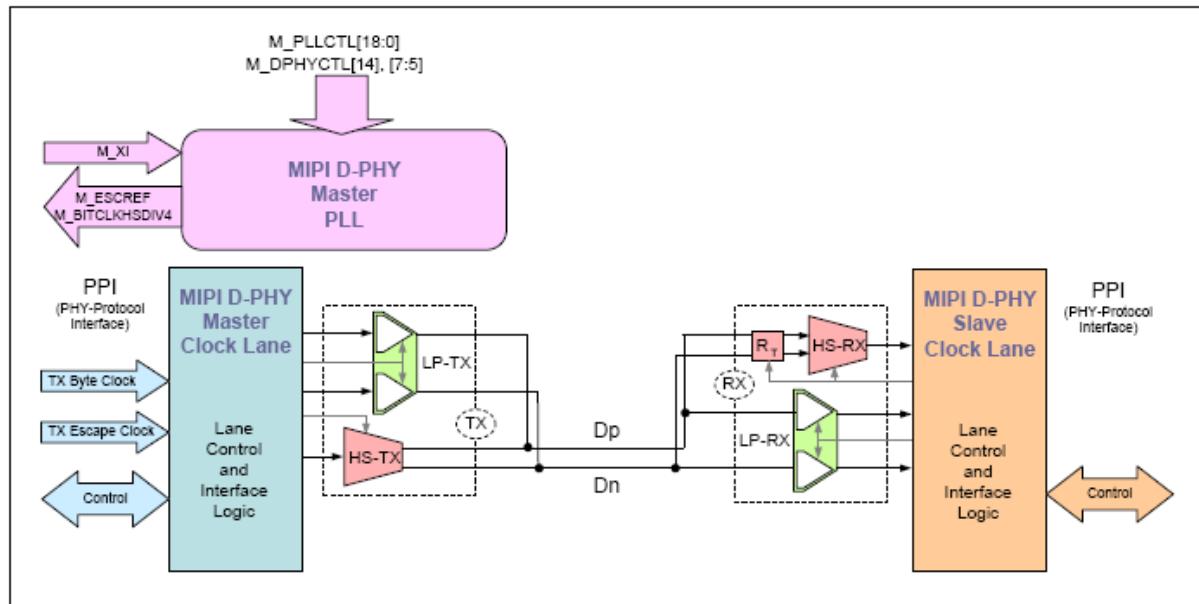


Figure 38-21. PLL and Clock Lane Connection

38.4.1.2 Data Lane Connection

The following figure illustrates the Data Lane connection.

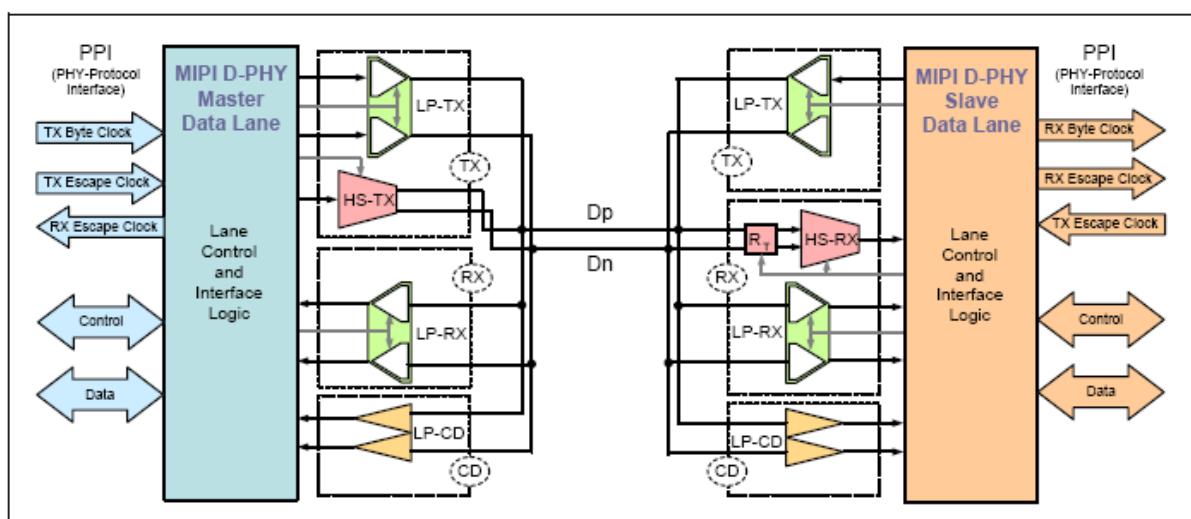


Figure 38-22. Data Lane Connection

38.4.1.3 IP Structure

The MIPI D-PHY core consists of five modules. The modules are:

- Master PLL
- Master Clock lane
- Master Data lane
- Slave Clock lane
- Slave Data lane

You can configure the modules depending on customer requirements.

For example:

- Expanding the number of data lanes up to four lanes
- Omitting the Master PLL when using another PLL for serial clock source
- Providing core in Hard Macro including IO and Power Pads.

We strongly recommend using Master and Slave lane in pair for loop-back test.

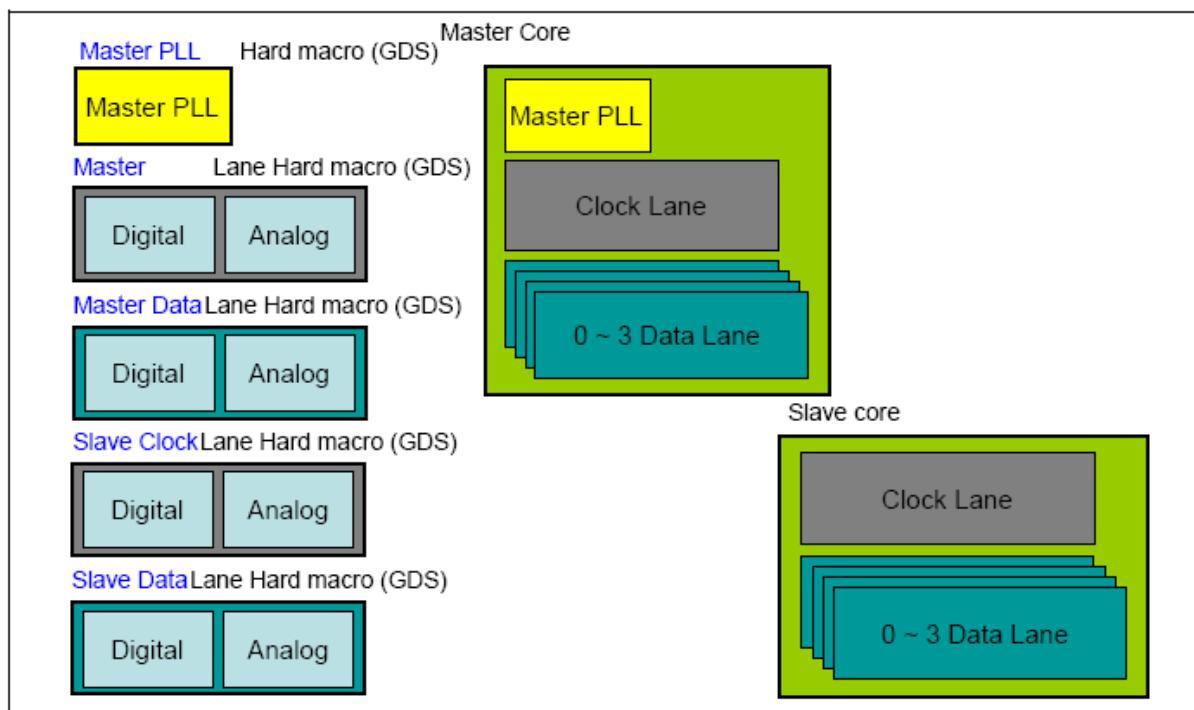


Figure 38-23. IP Structure

38.4.1.4 Power Consumption

- LN28LPP Process (without PAD)
- Operation Voltage conditions = 1.0 V±5 %, 1.8 V±5 %
- Recommended Operation Temperature range = -20°C to 85°C
- Master and slave lanes are assumed to have 1 clock + 1 data lane

The following table describes the simulation result >> condition : FF, -20°C, 1.05V, 1.89V

Mode	Master PLL	Master Lane	Slave Lane
HS (500 MHz)	2.8 mW	9 mW	3 mW
HS (1 GHz)	3.3 mW	12 mW	3.3 mW
LP (20 MHz)	-	1.2 mW	0.7 mW
ULPS	20 μ W	20 μ W	20 μ W

Mode	AVDD10	AVDD10_PLL	AVDD18
HS (500 MHz)	12.8 mW	2.8 mW	0.3 mW
HS (1 GHz)	16 mW	3.3 mW	0.3 mW
LP (20 MHz)	2.8 mW	-	2 mW

The following table describes the simulation result >> condition : NN, 25°C, 1.0V, 1.8V

Mode	Master PLL	Master Lane	Slave Lane
HS (500 MHz)	2.5 mW	7.5 mW	2.1 mW
HS (1 GHz)	3 mW	10.3 mW	2.4 mW
LP (20 MHz)	-	0.6 mW	0.4 mW
ULPS	10 μ W	10 μ W	10 μ W

Mode	AVDD10	AVDD10_PLL	AVDD18
HS (500 MHz)	10.2 mW	2.5 mW	0.2 mW
HS (1 GHz)	13.5 mW	3 mW	0.2 mW
LP (20 MHz)	1.8 mW	-	1.3 mW

38.4.1.5 PAD Signals

The following table describes the Pad signals.

Name	Description
M_VREG_0P4V	Regulator capacitor connection
M_DPCLK	Master CLK Lane DP
M_DNCLK	Master CLK Lane DN
M_DNDATA0/1/2/3	Master DATA Lanes DP
M_DNDATA0/1/2/3	Master DATA Lanes DN
S_DPCLK	Slave CLK Lane DP
S_DNCLK	Slave CLK Lane DN
S_DPDATA0/1/2/3	Slave DATA Lanes DP
S_DNDATA0/1/2/3	Slave DATA Lanes DN
M_VDD10_PLL	1.0 V Power for PLL
MS_VDD10	1.0 V Power for Internal Logic
MS_VDD18	1.8 V Power for Analog
MS_VSS	Ground
VREG12_EXTPWR	External 1.2 V power connection port. This is not a pad.

	If you use the Internal 1.2 V Regulator, VREG12_EXTPWR power port should float. Refer to description of B_DPHYCTL[20] of DSIM for mode setting information.
--	---

38.4.1.6 Package and Board Connection Guideline

Package and Board Connection Guideline section provides implementation information regarding package and board connections of MIPI D-PHY.

The following table describes the package and board requirements.

Signal Name	Description	Bonding Pad Requirements
M_VREG_0P4V	Analog Signal	Connect a 2nF capacitor between this pin and MS_VSS. Place capacitor closely to chip.
M_DPCLK M_DNCLK M_DPDATA0/1/2/3 M_DNDATA0/1/2/3 S_DPCLK S_DNCLK S_DPDATA0/1/2/3 S_DNDATA0/1/2/3	Analog Signal	The peak current through DP and DN pads is 5.5 mA (0.44 V/80 ohms), for a maximum duty cycle of 4 % during a short-to-ground condition. Make the total resistance of the package, ESD pad, and pad-macro connection 0.5 to 1.0 Ω . To reduce inductance and via in an array-type package, route these signals through shortest traces that reach outer contacts of array. Total capacitance of package, ESD pad, and pad-macro connection should be < 2 pF. Match delays of trace lines as close as possible to minimize skew between CLK's and DATA's. Require adjacent ball assignment to minimize intra-pair differential signals skew. Match CLK and DATA pairs for routing the path to minimize intra-pair differential signals skew. Therefore, you should consider symmetrical allocation for CLK and DATA pairs.
M_VDD10_PLL	PLL 1.0 V Supply	Capable of handling 10 mA. Make the resistance from the I/O pad (s) to the macro < 0.5 Ω , with < 5 % voltage variation at the macro. The VDD and VSS connections require dedicated off-chip supplies.
MS_VDD10	D-PHY 1.0 V Supply	Capable of handling 5 mA per Lane. Make the resistance from the I/O pad (s) to the macro < 0.5 Ω , with < 5 % voltage variation at the macro. The VDD and VSS connections require dedicated off-chip supplies.
MS_VDD18	D-PHY 1.8 V Supply	Capable of handling 2 mA. Make the resistance from the I/O pad (s) to the macro < 0.5 Ω , with < 10 % voltage variation at the macro. The VDD and VSS connections require dedicated off-chip supplies.
MS_VSS	Ground	Common Ground

The followings are RC Guide Line for PKG.

- Differential Signals: $R < 0.5 \Omega$ and $C < 1 \text{ pF}$
- Power Signals: $R < 1 \Omega$ and $C < 1 \text{ pF}$
- You should consider bump arrangement and package PCB design to ensure high speed differential pair signals goes out to last outer ball (PKG).
- CLK/DATA pairs should have the same routing length (difference < 20um).
- PAD to BUMP metal line should be straight as possible.
- You should not place the other signals which does not relate to MIPI near MIPI differential signals. ($S/W > 10$, S: Line Space, W: Line Width)

38.4.1.7 Core Interface Timing Diagram

Clock Lane: HS-TX and HS-RX function

The following feature illustrates the HS-TX and HS-RX function of Clock Lane.

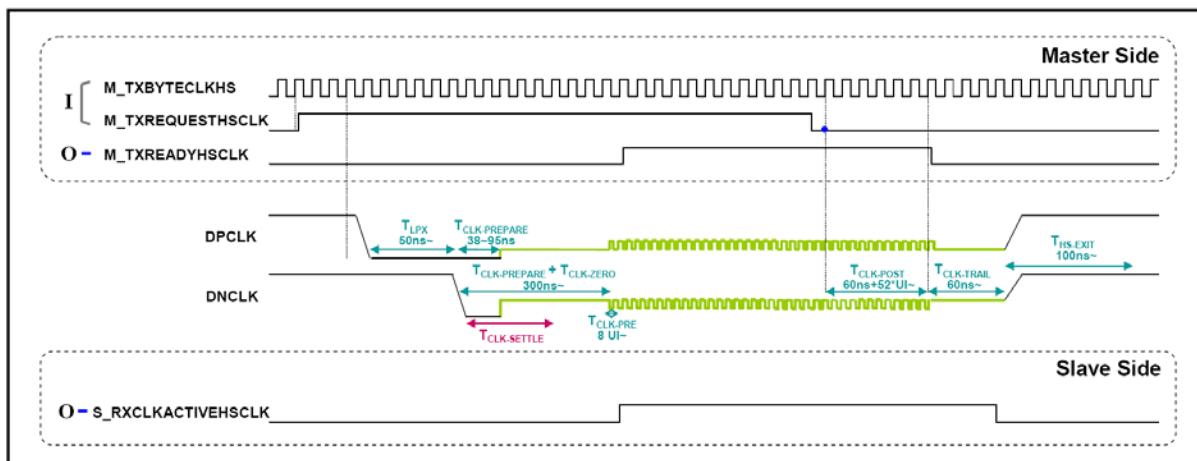


Figure 38-24. Clock Lane: HS-TX and HS-RX function

Data Lane: HS-TX and HS-RX Function

The following feature illustrates the HS-TX and HS-RX function of Data Lane.

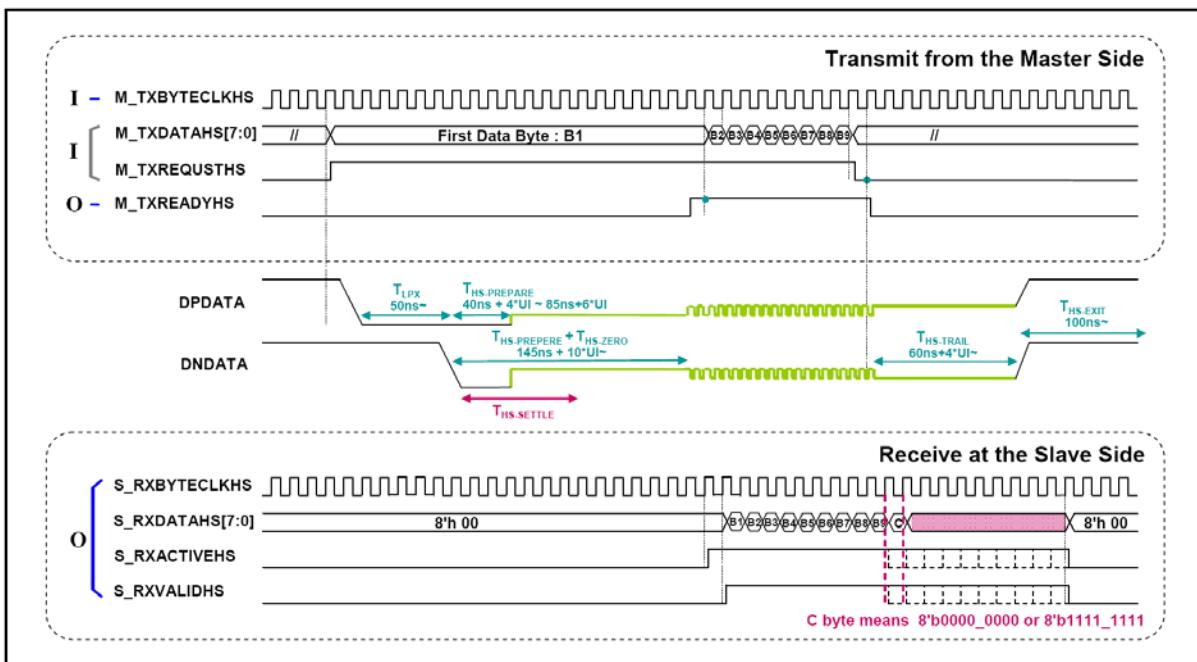


Figure 38-25. Data Lane: HS-TX and HS-RX Function

Clock Lane: ULPS Function

The following feature illustrates the ULPS function of Clock Lane.

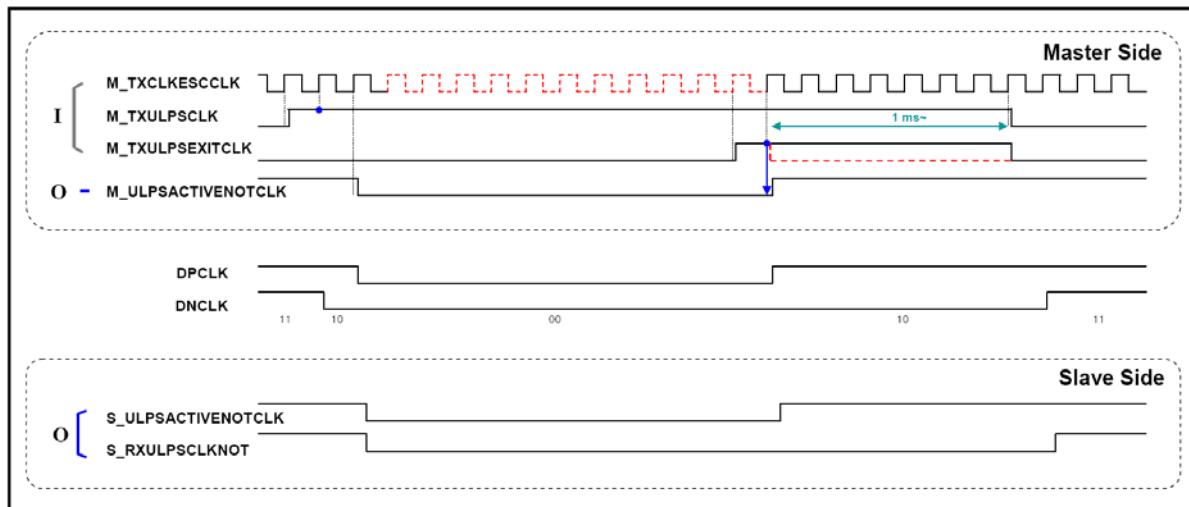


Figure 38-26. Clock Lane: ULPS Function

Data Lane: ULPS Function

The following feature illustrates the ULPS function of Data Lane.

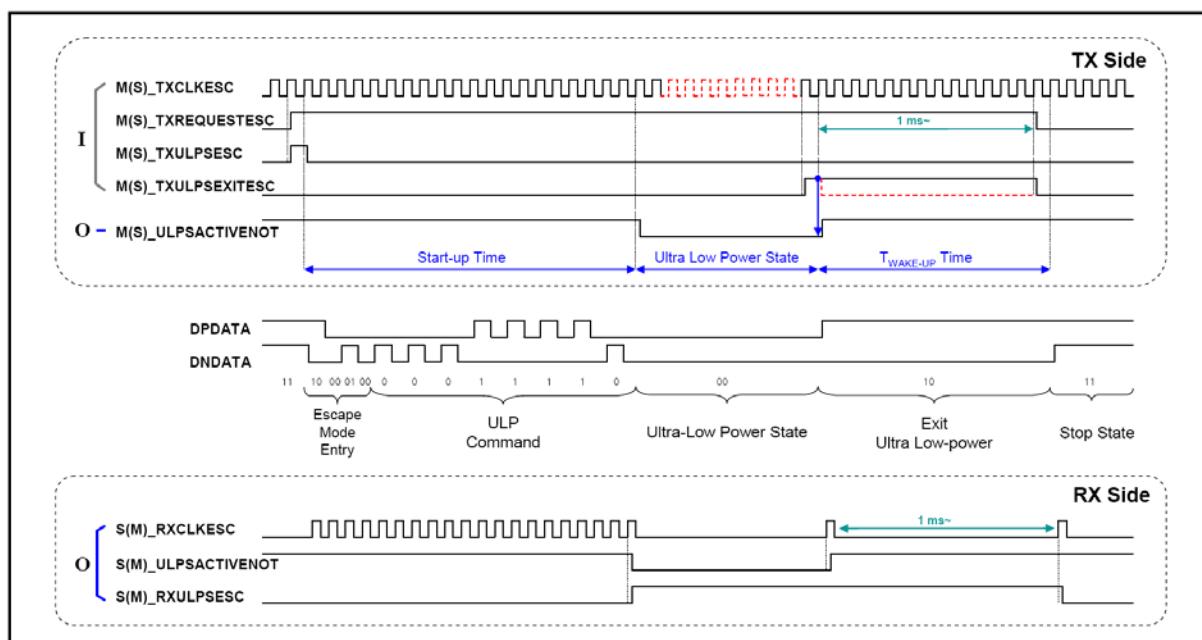


Figure 38-27. Data Lane: ULPS Function

Data Lane: LP-TX and LP-RX Function

The following feature illustrates the LP-TX and LP-RX function of Data Lane.

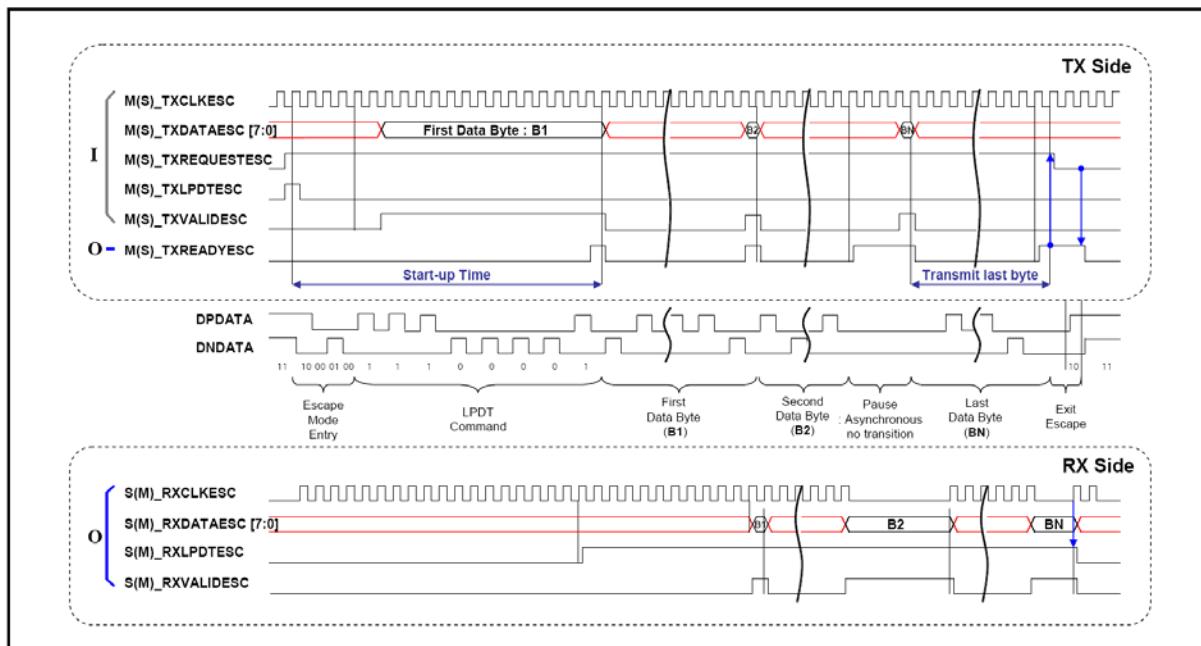


Figure 38-28. Data Lane: LP-TX and LP-RX Function

Data Lane: Remote Trigger Reset

The following feature illustrates the remote trigger reset of Data Lane.

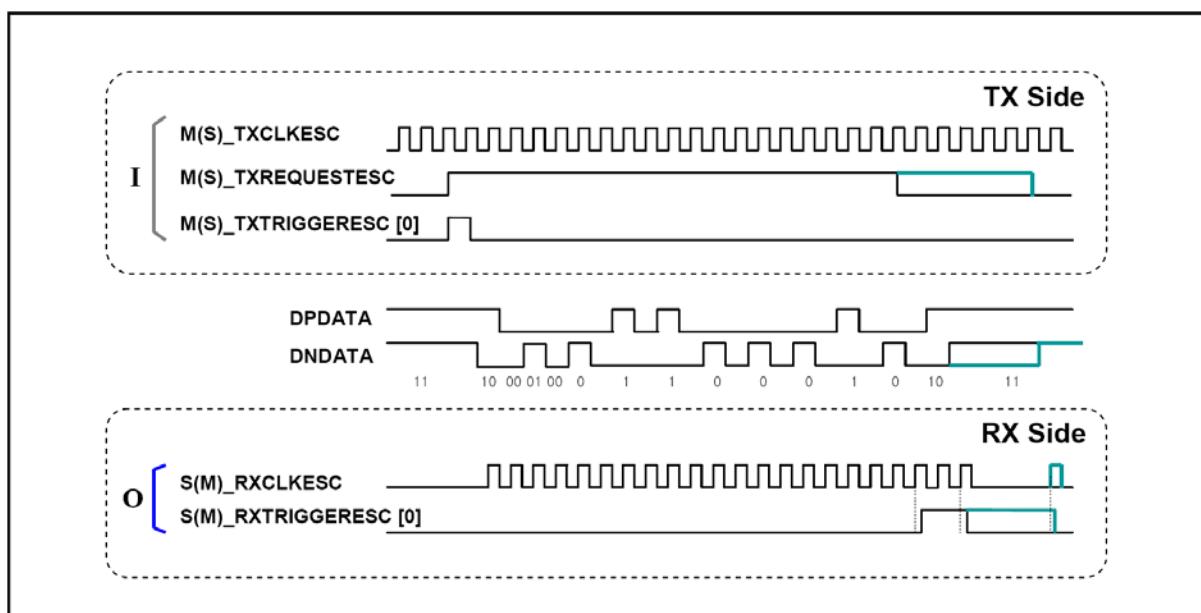


Figure 38-29. Data Lane: Remote Trigger Reset

Data Lane: Turn Around

The following feature illustrates the turn around of Data Lane.

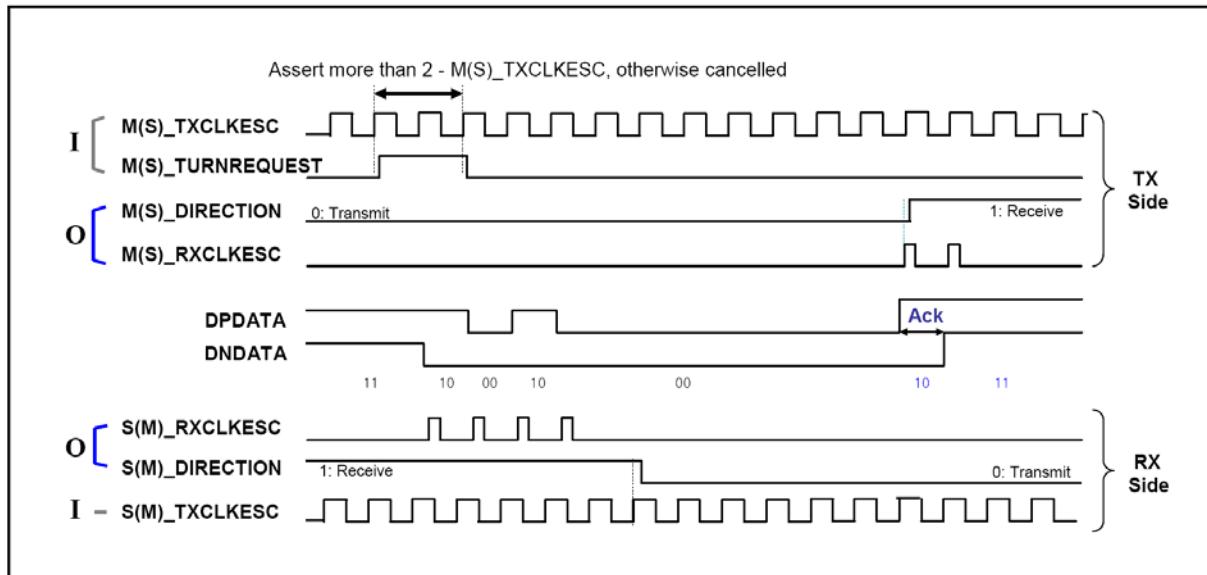


Figure 38-30. Data Lane: Turn Around

Initialization Sequence

The following feature illustrates the initialization sequence.

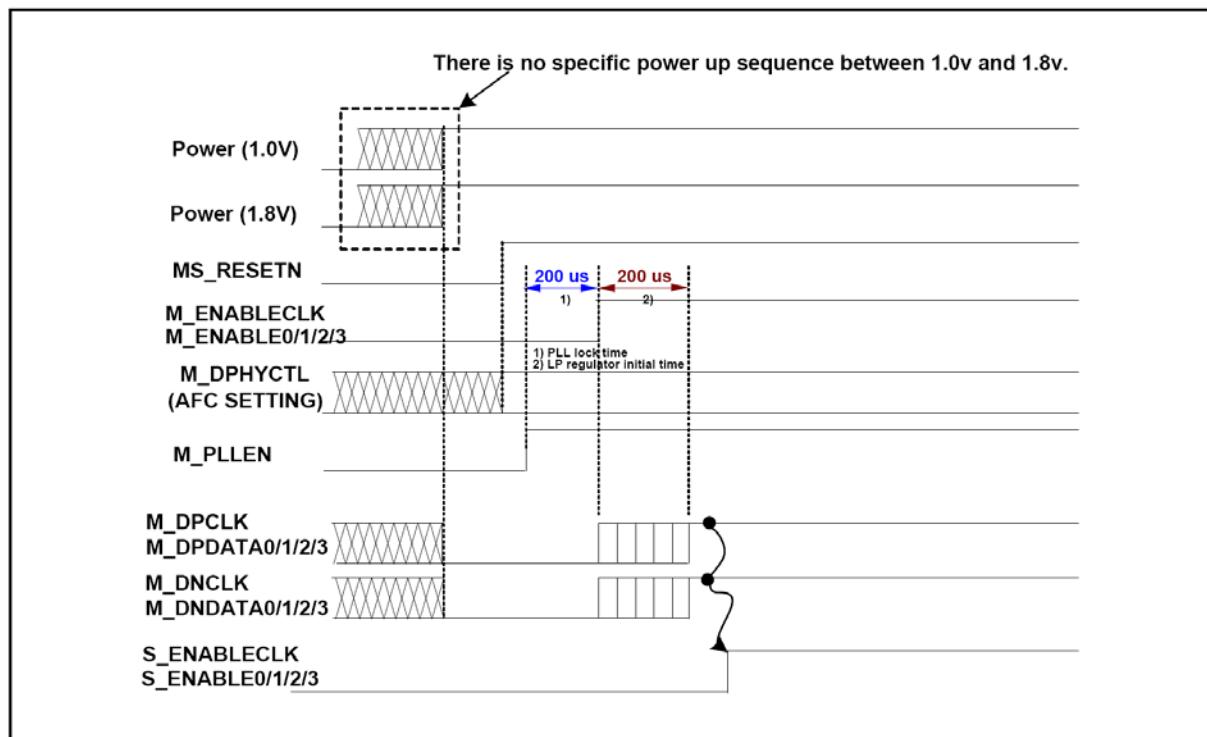


Figure 38-31. Initialization Sequence

Section 39. Video Input Processor (VIP)

39.1 Overview

The Video Input Processor (hereinafter VIP) of the NXP4330D/Q can receive images directly from external camera modules or video decoders. In addition, it can clip or scale down the input images and store them to the memory. The images stored from the VIP can be used for encoding by using MPEG Hardware, and as preview images by using the Multi-Layer Controller (MLC). In addition, the images can be converted to texture images for the 3D Graphics Accelerator by using the Color Space Converter.

39.1.1 Features

The Video Input Port features:

- ITU-R BT.656(External CIS) and ITU-R BT.601 (External CIS and MIPI, External 8-bit, MIPI 16bit) interface supports
- Clock, HSYNC, VSYNC and 8-bit data port (External CIS)
- Clock, HVALID, VVALID, DVALID and 16bit data port (Internal MIPI CSI)
- External DVALID pin or Field pin supports
- Maximum 8192 x 8192 image supports
- Clipping and Scale-down
- Horizontal Flip Effects
- YUV 420/422/444 memory format and Linear YUV 422 memory format
- Horizontal & Vertical Interrupt and Operation Done Interrupt
- Internal Decoder and External interface supports
- 3 Inputs from External CIS and 1 Input from MIPI CSI

39.2 VIP Interconnection

39.2.1 Block Diagram

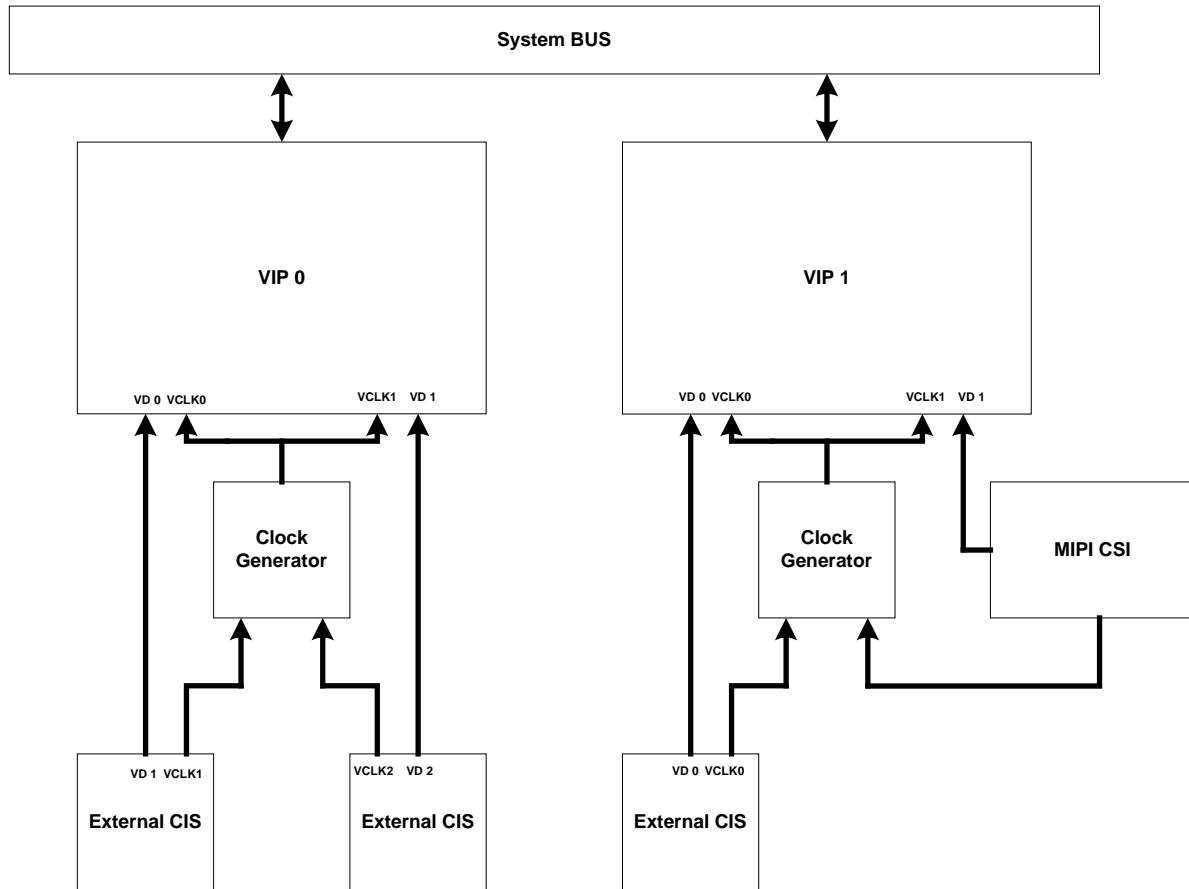


Figure 39-1. Video Input Port Interconnection

The VIP Interconnected with 3 External CIS modules and 1 MIPI CSI module as shown as Figure 39-1.

39.2.2 Clock Generation

The VIP can create the video in clock by using an internal PLL or an external VCLK pin as a clock source. The created video in clock is used for sync signal creation and data interface in the Video Input Port block. In general, the Video Input port is designed to fetch data on the rising edge. Therefore, if the video clock fetches data at the falling edge, the **CLKSRCSEL** should be set as '4'.

39.3 Video Input Port

39.3.1 Block Diagram

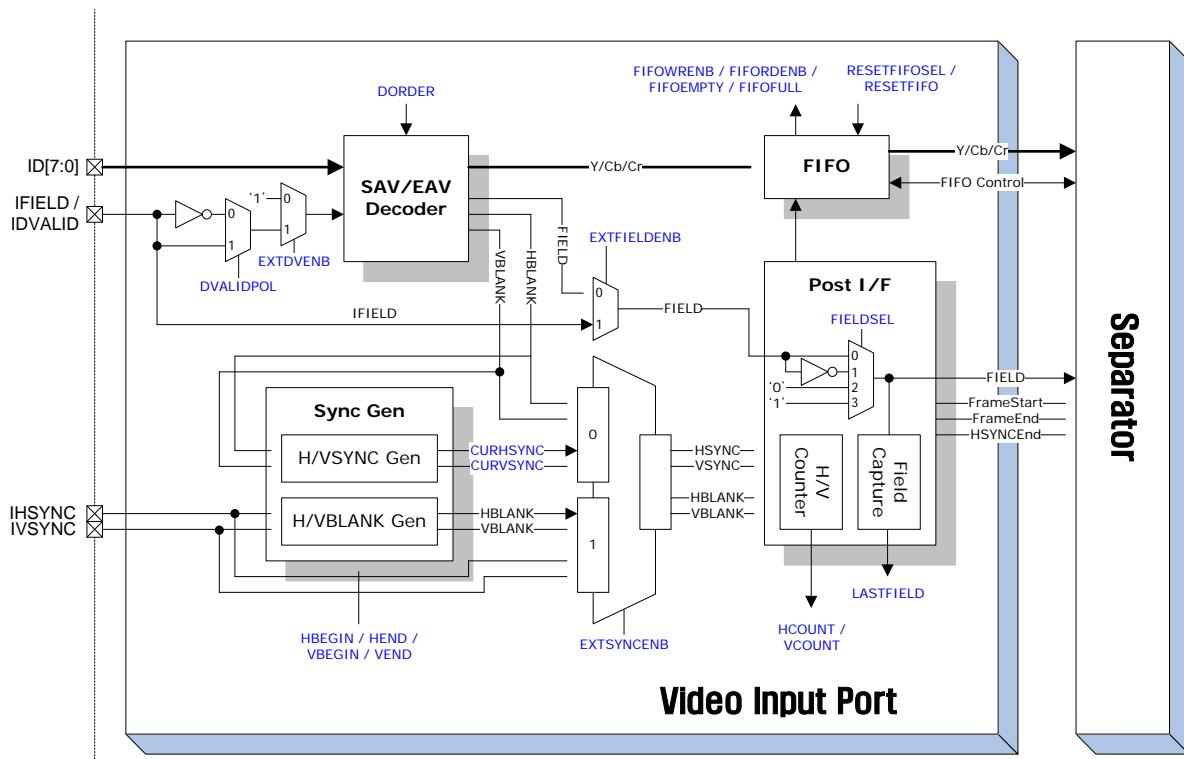


Figure 39-2. Video Input Port Block Diagram

39.3.2 Sync Generation

The horizontal and vertical timing interfaces for the video input port are as shown in Figure 39-3.

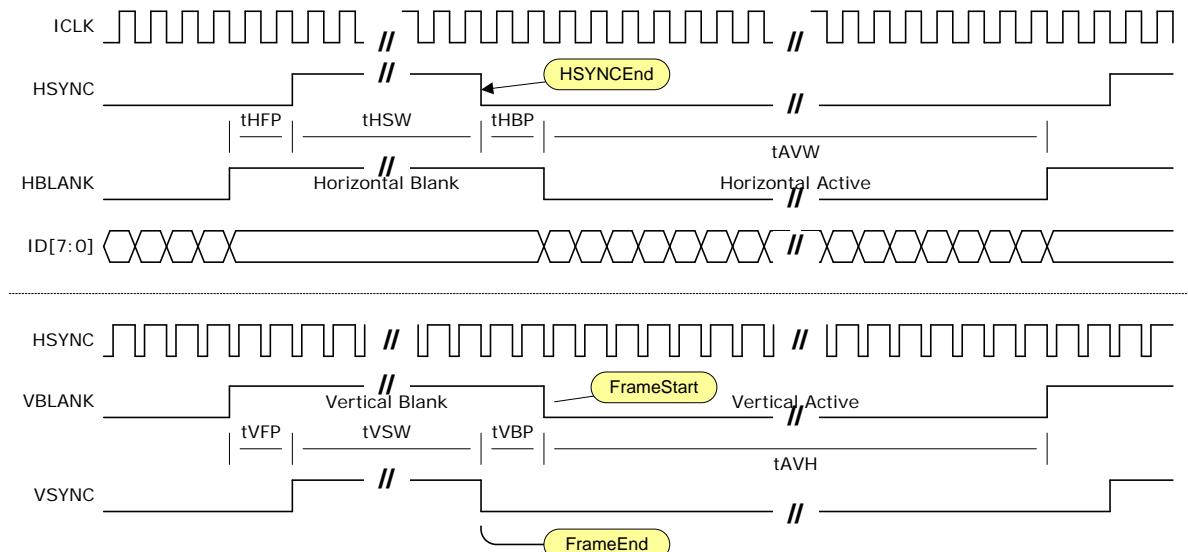


Figure 39-3. Horizontal & Vertical Timings

Each symbol in Figure 39-3 is described in Table 39-1.

Symbol	Brief	Remark
tHSW	Horizontal Sync Width	Number of ICLKs in the section where the horizontal sync pulse is active
tHBP	Horizontal Back Porch	Number of ICLKs in a section from the end point of the horizontal sync pulse to the start point of the horizontal active
tHFP	Horizontal Front Porch	Number of ICLKs in a section from the end point of the horizontal active to the start point of the horizontal sync pulse
tAVW	Active Video Width	Number of ICLKs in a horizontal active section
tVSW	Vertical Sync Width	Number of lines in a section where the vertical sync pulse is active
tVBP	Vertical Back Porch	Number of lines in a section from the end point of the vertical sync pulse to the start point of the next vertical active
tVFP	Vertical Front Porch	Number of lines in a section from the end point of the vertical active to the start point of the vertical sync pulse
tAVH	Active Video Height	Number of lines in the vertical active section

Table 39-1. Horizontal & Vertical timing symbols

ITU-R BT.601 8-bit

The video input port has an 8-bit data bus and HSYNC and VSYNC pins, and supports ITU-R BT.601 8-bit input. If the port uses an external HSYNC or VSYNC, the **EXTSYNCENB** bit should be set as '1'. If the **EXTSYNCENB** bit is '1', the port receives the HSYNC and VSYNC from the outside, and creates the HBLANK and VBLANK internally. The polarity of the external H(V)SYNC only supports high active. Figure 39-4 shows the relationship between the HBLANK and the VBLANK generated from external HSYNC and VSYNC inputs.

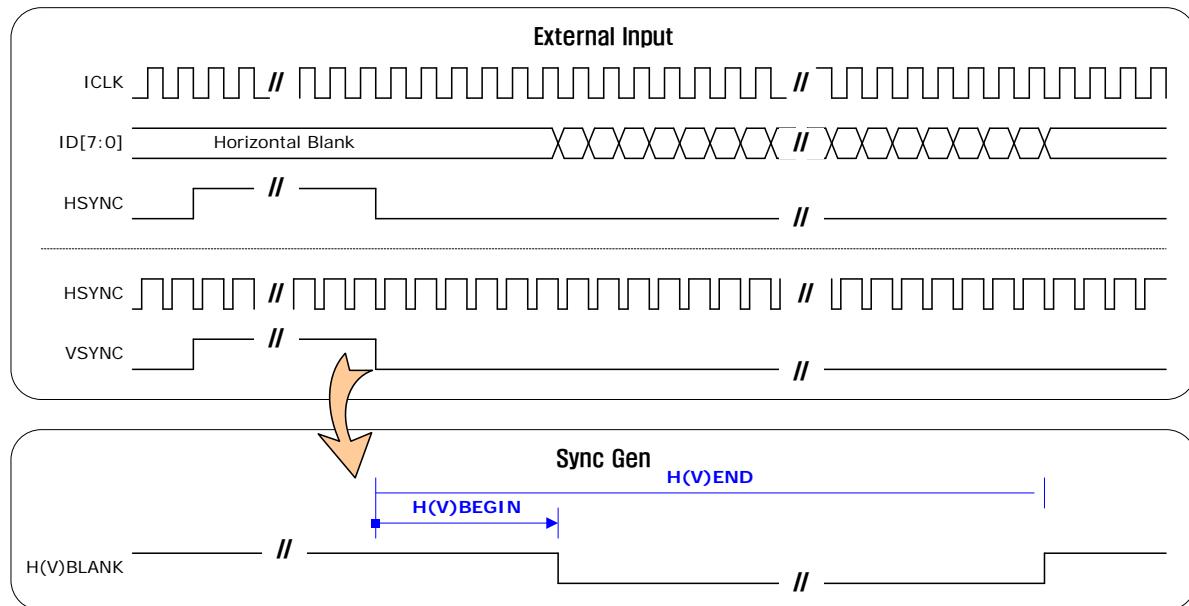


Figure 39-4. Sync Generation for ITU-R BT.601 8-bit

The **HBEGIN** and the **HEND** are used to generate the HBLANK from the HSYNC. The **VBEGIN** and the **VEND** are used to generate the VBLANK from the VSYNC. The settings for each register are listed in Table 39-2.

Register	Formula	Remark
VBEGIN	$tVBP - 1$	Number of lines in a section from the end point of the VSYNC to the start point of the vertical active video - 1
VEND	$tVBP + tAVH - 1$	Number of lines in a section from the end point of the VSYNC to the end point of the vertical active video - 1
HBEGIN	$tHBP - 1$	Number of clocks in a section from the end point of the HSYNC to the start point of the horizontal active video - 1
HEND	$tHBP + tAVW - 1$	Number of clocks in a section from the end point of the HSYNC to the start point of the horizontal active video - 1

Table 39-2. Register settings for ITU-R BT.601 8-bit

ITU-R BT.656

In the ITU-R BT.656 format, there is no additional sync signal pin, and the sync information is transmitted along with data via data pins. At this time, the Sync information is inserted as an additional code before the start point of the valid data (SAV) and after the end of the valid data (EAV). Sync information included in data is as shown in Figure 39-5.



Figure 39-5. Data stream format with SAV/EAV

The SAV and the EAV consists of [FF, 00, 00, CODE]. Each code contains Field (F), VSYNC (V) and HSYNC (H) data, and each code is composed as follows:

Bit	7	6	5	4	3	2	1	0	Hex	Brief Description
Function	1	F	V	H	P3	P2	P1	P0		
(FVH)	0	1	0	0	0	0	0	0	80h	SAV of odd field
	1	1	0	0	1	1	1	0	9Dh	EAV of odd field
	2	1	0	1	0	1	0	1	ABh	SAV of odd blank
	3	1	0	1	1	0	1	1	B6h	EAV of odd blank
	4	1	1	0	0	0	1	1	C7h	SAV of even field
	5	1	1	0	1	1	0	1	DAh	EAV of even field

	6	1	1	1	0	1	1	0	0	ECh	SAV of even blank
	7	1	1	1	1	0	0	0	1	F1h	EAV of even blank
▪ F : Field select (0 : odd field, 1 : even field)											
▪ V : Vertical blanking (0 : Active, 1 : blank)											
▪ H : SAV/EAV (0 : SAV, 1 : EAV)											
▪ Parity : P3 = V xor H, P2 = F xor H, P1 = F xor V, P0 = F xor V xor H											

Table 39-3. Embedded Sync Code

To create the HBLANK and VBLANK from the sync information contained in the data, the **EXTSYNCENB** should be set as '0'. The SAV/EAV decoder blocks generate the HBLANK and VBLANK from the sync information contained in the data. The Sync Gen block generates the HSYNC and the VSYNC based on the HBLANK and VBLANK signals. Figure 39-6 shows the relationship that generates the H(V)BLANK and H(V)SYNC from the SAV/EAV contained in the data.

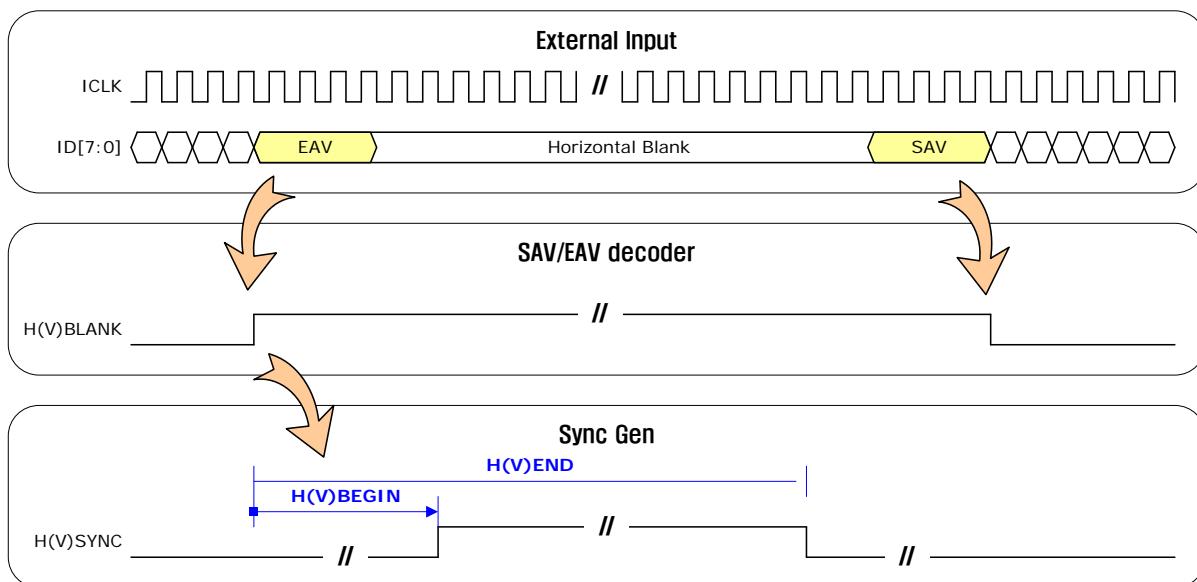


Figure 39-6. Sync Generation for ITU-R BT.656

If the **EXTSYNCENB** is '0', the **HBEGIN** and **HEND** are used to generate the HSYNC signal from the HBLANK. The **VBEGIN** and **VEND** are used to generate the VSYNC signal from the VBLANK. The settings for each register are listed in Table 39-4.

Register	Formula	Remark
VBEGIN	tVFP + 1	Number of lines in a section from the end point of the vertical active video to the start point of the VSYNC - 1
VEND	tVFP + tVSW + 1	Number of lines in a section from the end point of the vertical active video to the end point of the VSYNC + 1
HBEGIN	tHFP - 7	Number of clocks in a section from the end point of the horizontal active video to the start point of the HSYNC - 7
HEND	tHFP + tHSW - 7	Number of clocks in a section from the end point of the horizontal active video to the end point of the HSYNC - 7

Table 39-4. Register settings for ITU-R BT.656

ITU BT.656-like support

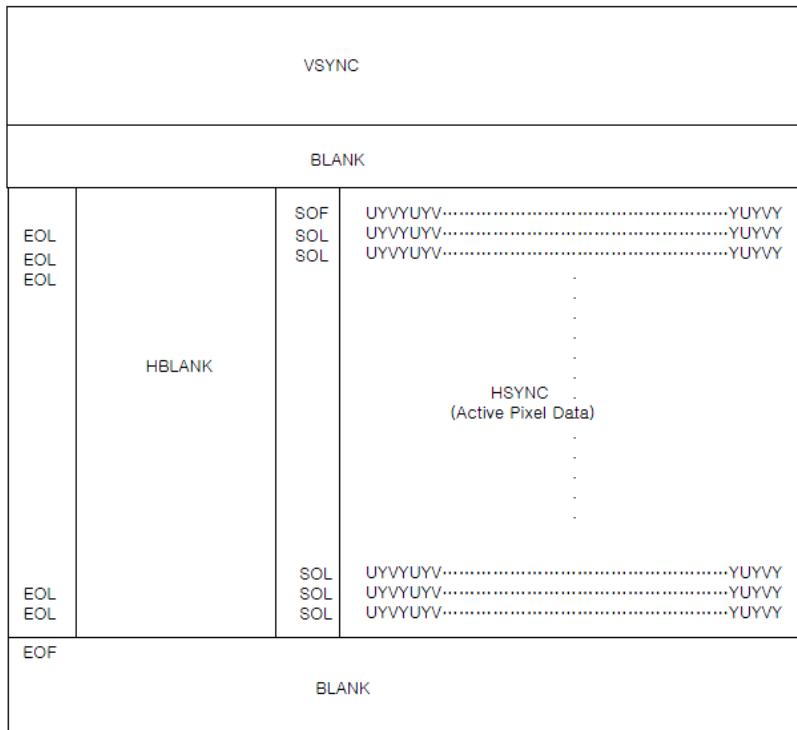


Figure 39-7. Frame structure of ITU656-like

Our VIP supports ITU656-like which could be configurable as follows:

- SOF = 0xFF00_0080
- EOF = 0xFF00_00B6
- SOL = 0xFF00_0080
- EOL = 0xFF00_009D

39.3.3 External Data Valid and Field

The video input port can receive data valid signals or field signals from the outside. Since the IDVALID and the IFIELD share a pin, users can use only one of them.

External Data Valid

If the ***EXTDVENB*** is set as '1', users can use the input signal from the IFIELD/IDVALID pad as the IDVALID signal. In this case, the polarity is determined by the ***DVALIDPOL***. The video input port is designed to use the IDVALID signal of active high mode internally. Therefore, if the polarity of an external IDVALID signal is active low, the input signal should be inverted by setting the ***DVALIDPOL*** as '0'. If the polarity of an external IDVALID signal is active high, the input signal should be bypassed by setting the ***DVALIDPOL*** as '1'.

Even though an external IDVALID signal is used, the internal HBLANK and VBLANK signals are used. Therefore, the user should set the ***H(V)BEGIN*** and the ***H(V)END***.

External Field

If an external field signal is used, the ***EXTFIELDENB*** should be set as '1'. In the ITU-R BT.656 format, the input signal from the IFIELD/IDVALID pad can be used as an external field signal by setting the ***EXTFIELDENB*** as '1'. The video input port internally considers it as an odd field if the polarity of the field signal is low. If the polarity of a field signal is high, the port considers it as an even field. The user can select the polarity of a field signal by using ***FIELDSEL***, or can

fix the polarity as '0' or '1'.

39.3.4 Data Order

The video input port can select the order of input data. Basically, the ITU-R BT.656 format or the ITU-R BT.601 8-bit format has the order [Cb, Y0, Cr, Y1]. If the order of the input data is different from the default order, users can change the order via **DORDER**. The data orders supported by the video input port are listed in Table 39-5.

DORDER	0				1				2				3			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
ICLK	Cb[7]	Y _N [7]	Cr[7]	Y _{N+1} [7]	Cr[7]	Y _{N+1} [7]	Cb[7]	Y _N [7]	Y _N [7]	Cb[7]	Y _{N+1} [7]	Cr[7]	Y _{N+1} [7]	Cr[7]	Y _N [7]	Cb[7]
ID[7]	Cb[6]	Y _N [6]	Cr[6]	Y _{N+1} [6]	Cr[6]	Y _{N+1} [6]	Cb[6]	Y _N [6]	Y _N [6]	Cb[6]	Y _{N+1} [6]	Cr[6]	Y _{N+1} [6]	Cr[6]	Y _N [6]	Cb[6]
ID[5]	Cb[5]	Y _N [5]	Cr[5]	Y _{N+1} [5]	Cr[5]	Y _{N+1} [5]	Cb[5]	Y _N [5]	Y _N [5]	Cb[5]	Y _{N+1} [5]	Cr[5]	Y _{N+1} [5]	Cr[5]	Y _N [5]	Cb[5]
ID[4]	Cb[4]	Y _N [4]	Cr[4]	Y _{N+1} [4]	Cr[4]	Y _{N+1} [4]	Cb[4]	Y _N [4]	Y _N [4]	Cb[4]	Y _{N+1} [4]	Cr[4]	Y _{N+1} [4]	Cr[4]	Y _N [4]	Cb[4]
ID[3]	Cb[3]	Y _N [3]	Cr[3]	Y _{N+1} [3]	Cr[3]	Y _{N+1} [3]	Cb[3]	Y _N [3]	Y _N [3]	Cb[3]	Y _{N+1} [3]	Cr[3]	Y _{N+1} [3]	Cr[3]	Y _N [3]	Cb[3]
ID[2]	Cb[2]	Y _N [2]	Cr[2]	Y _{N+1} [2]	Cr[2]	Y _{N+1} [2]	Cb[2]	Y _N [2]	Y _N [2]	Cb[2]	Y _{N+1} [2]	Cr[2]	Y _{N+1} [2]	Cr[2]	Y _N [2]	Cb[2]
ID[1]	Cb[1]	Y _N [1]	Cr[1]	Y _{N+1} [1]	Cr[1]	Y _{N+1} [1]	Cb[1]	Y _N [1]	Y _N [1]	Cb[1]	Y _{N+1} [1]	Cr[1]	Y _{N+1} [1]	Cr[1]	Y _N [1]	Cb[1]
ID[0]	Cb[0]	Y _N [0]	Cr[0]	Y _{N+1} [0]	Cr[0]	Y _{N+1} [0]	Cb[0]	Y _N [0]	Y _N [0]	Cb[0]	Y _{N+1} [0]	Cr[0]	Y _{N+1} [0]	Cr[0]	Y _N [0]	Cb[0]

Table 39-5. Input Data Order

39.3.5 Status

Horizontal & Vertical Counter

The video input port can inform the user of the size of the active section. The **HCOUNT** indicates the total clock numbers of a line in the active video section. The **VCOUNT** indicates the total line numbers in the active video section.

Current HSYNC & VSYNC Status

When the **EXTSYNCENB** is '0', the **CURHSYNC** and the **CURVSYNC** bits are provided to show the HSYNC and VSYNC states.

Current Field Status

Users can display the status of the current field signal by using the **LASTFIELD** bit as shown in Figure 39-8.

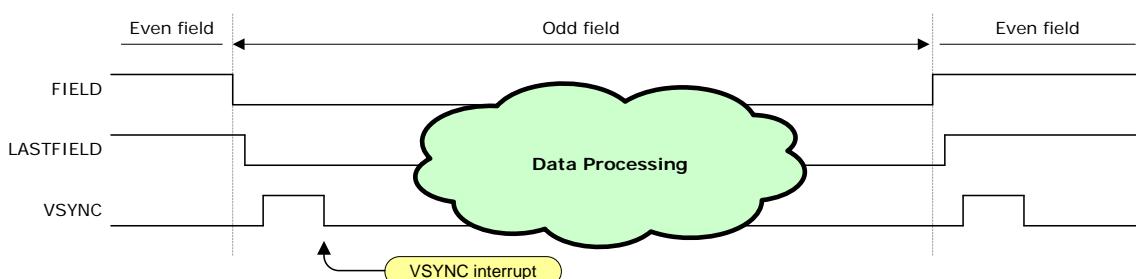


Figure 39-8. Field information

LASTFIELD is updated whenever a field signal is changed. In addition, when a VSYNC interrupt occurs, the user can get the field status of the next data by using **LASTFIELD**. Since **LASTFIELD** is updated by using the PCLK, the PCLK should be always enabled by setting the **PCLKMODE** as '1' to obtain a proper field status.

39.3.6 FIFO Controls

The video input port block can inform the user of the current status of the internal FIFO. The **FIFOWRENB** indicates if data is being written to the FIFO. The **FIFORDENB** indicates if data is being read from the FIFO. If the FIFO is empty, the **FIFOEMPTY** is set as '1'. If the FIFO is full, the **FIFOFULL** is set as '1'.

In addition, users can reset the FIFO at a specific point. According to the **RESETIFOSEL** setting, the reset point of the FIFO can be controlled by selecting either FrameEnd (the end of VSYNC, **RESETIFOSEL** is '0'), FrameStart (the start of vertical active video, **RESETIFOSEL** is '1') or the **RESETIFO** bit (**RESETIFOSEL** is '2'), or by selecting all of them (**RESETIFOSEL** is '3'). The **RESETIFO** bit is valid only when the **RESETIFOSEL** is '2' or '3'. If the FIFO is reset by setting the **RESETIFO** as '1', the **RESETIFO** should be set as '0' again.

39.3.7 Recommend Setting for Video Input Port

Table 39-6 lists the recommend settings for the video input port by input formats.

Register	ITU-R BT.656	ITU-R BT.601 8-bit	
		Progressive	Interlace
EXTSYNCENB	0	1	
DWIDTH	1	1	
DORDER	0 (default)	0 (default)	
EXTFIELDENB	0	0	1
FIELDSEL	0	3	0 or 1
EXTDVENB	0	0 or 1	0
DVALIDPOL	Not used	0 or 1	Not used
VBEGIN	tVFP + 1	tVBP - 1	
VEND	tVFP + tVSW + 1	tVBP + tAVH - 1	
HBEGIN	tHFP - 7	tHBP - 1	
HEND	tHFP + tHSW - 7	tHBP + tAWW - 1	

Table 39-6. Recommend setting for Video Input Port

39.4 CLIPPER & DECIMATOR

39.4.1 Clipping & Scale-down

The VIP can store input images to the memory after clipping or scaling down. An input image is transmitted to the Clipper through the video input port and the Separator. The Clipper clips a specific area from the input image, and then stores the result in the memory and transmits the result to the Decimator. The Decimator can store the image transmitted from the Clipper in the memory after scaling down the image. Figure 39-9 shows the procedure for clipping and scaling down an input image.

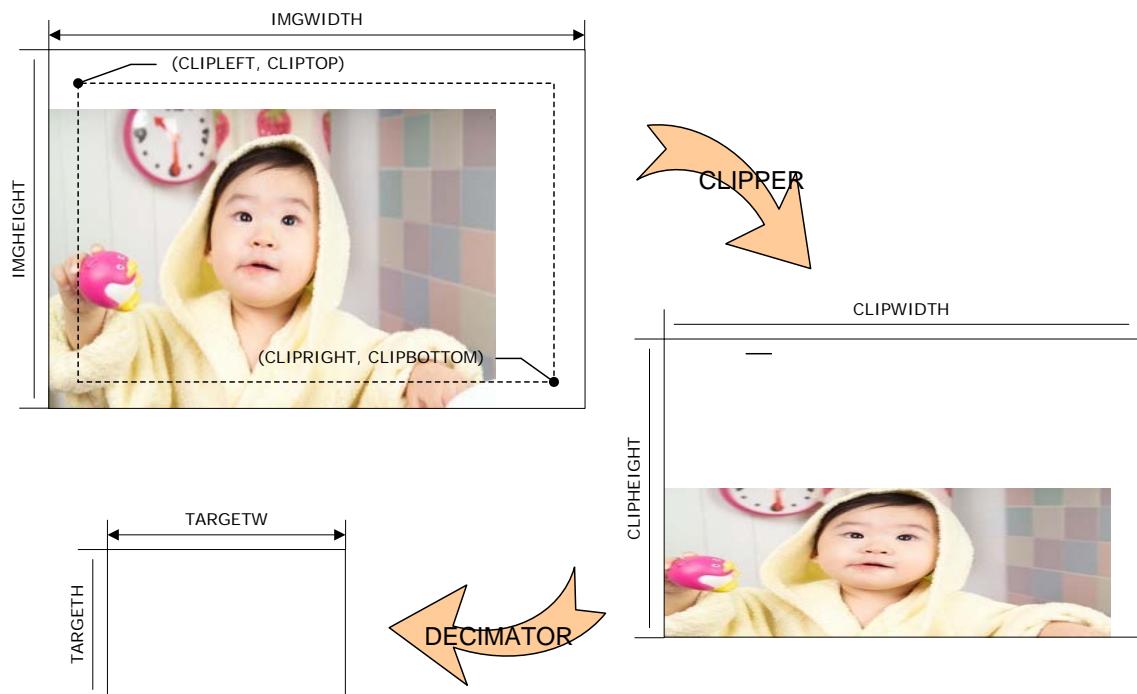


Figure 39-9. Clipping & Decimation

Users can enable the Clipper to clip an input image by specifying the relevant area, using ***CLIPLEFT***, ***CLIPRIGHT***, ***CLIPTOP*** and ***CLIPBOTTOM***.

The Decimator scales down the clipped image by using the Bresenham algorithm. To this end, ***TARGETW***, ***TARGETH***, ***DELTAW***, ***DELTAH***, ***CLEARW***, and ***CLEARH*** are used. Table 39-7 lists the settings for each register.

Register	Formula	Range	Unit	Remark
TARGETW	—	0 ~ 8191	Pixel	Width of a scaled-down image
TARGETH	—	0 ~ 8191	Pixel	Height of a scaled-down image
DELTAW	CLIPWIDTH - TARGETW	0 ~ 8191	Pixel	Width difference between the original image and the result image
DELTAH	CLIPHEIGHT - TARGETH	0 ~ 8191	Pixel	Height difference between the original image and the result image
CLEARW	TARGETW - DELTAW	-8192 ~ 8191	Pixel	Difference between the width of the result image and the DELTAW
CLEARH	TARGETH - DELTAH	-8192 ~ 8191	Pixel	Difference between the height of the result image and the DELTAH

Table 39-7. Registers for scaling

39.4.2 Output Data Format

The VIP can store input images in 2D block separated YUV format and linear YUV 4:2:2 format. The Clipper supports both 2D block separated YUV format and linear YUV 4:2:2 format. If the ***YUYVENB*** is set as '0', data is stored in 2D block separated YUV format. If the ***YUYVENB*** is set as '1', data is stored in linear YUV 4:2:2 format. The Decimator only supports 2D block separated YUV format.

Linear YUV 4:2:2 format

Linear YUV format is the YUYV format, and Y (luminance) exists in each pixel. Cb and Cr (Chrominance) separately exist in each of two pixels, and two pixels share the Cr and the Cb (Chrominance). The VIP has 2-pixel information per 32-bit and is managed in 2-pixel units. Table 39-8 shows the memory format that Y/Cb/Cr data are stored in.

Pixel format	YUYVENB	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YUYV	1					Cb[7:0]								Y1[7:0]							Cb[7:0]								Y0[7:0]				

Table 39-8. YUYV format

The ***BASEADDR(H)*** and the ***STRIDE(H)*** are used for the addressing of the linear YUV 4:2:2 format. The ***BASEADDR*** is a 32-bit linear address, and specifies the memory address where output images from the Clipper are stored. The ***STRIDE*** is the memory offset from one scan line in the image buffer to the next. The ***STRIDE*** is expressed in byte units and is also called pitch. If there is no-memory gap between lines in the image buffer, the stride can be specified as '***CLIPWIDTH*2***'.

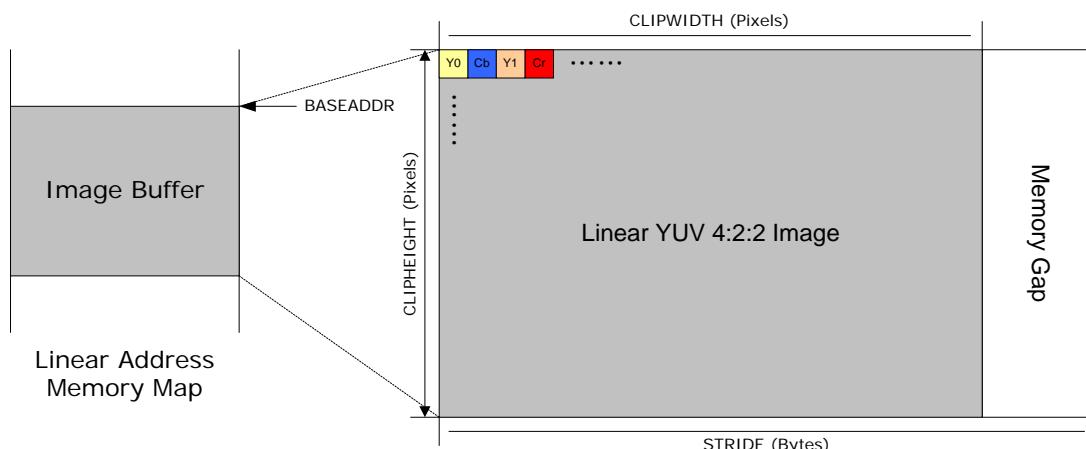


Figure 39-10. Address Generation for Linear YUV 422 Format

The linear YUV 4:2:2 format is only supported by the Clipper, and ***YUYVENB*** should be set as '1'.

2D Block Separated YUV format

In 2D block separated YUV format, each of Y, U and V exists at separate memory spaces. In addition, 2D block separated YUV format is divided into 4:4:4, 4:2:2 and 4:2:0 in proportion to U and V for Y. 2D block separated YUV format is the 2D block addressing format, and each component has a size of 64 x 32 and linearity in block units. These features provide the NXP3200's unique memory format, to enhance the effectiveness of memory access when the NXP3200 manages data in macro block units through an algorithm to compress/decompress images such as MPEG files.

According to each format, Y, U and V correspond to 2 x 2 pixels as follows:

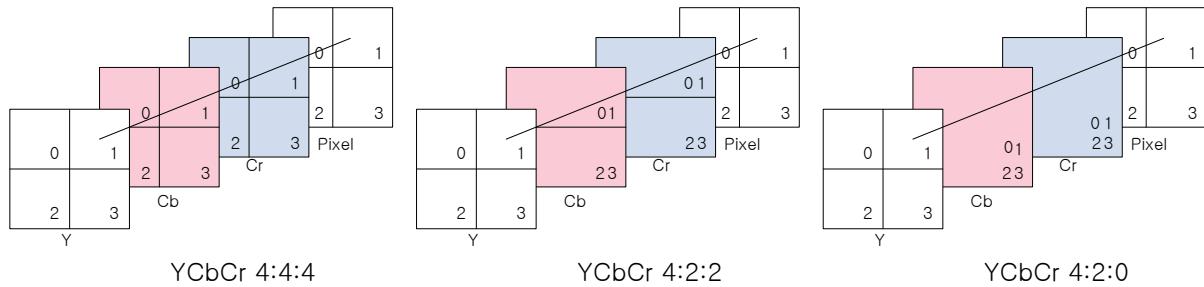


Figure 39-11. Separated YUV format

Users can select either 4:4:4, 4:2:2 or 4:2:0 format by using **FORMATSEL**. In the Clipper, if **YUYVENB** is set as '0', data is stored in the 2D block separated YUV format.

In the 2D block addressing format, the memory address to which data is stored is specified by using a segment and X/Y coordinates. A segment has 4096×4096 spaces and occupies 16MByte memory space. X and Y coordinates can have values between 0 and 4095 as the offsets for each axis in a segment. The X-axis area is specified by using LEFT and RIGHT, and the Y-axis area is specified by using TOP and BOTTOM. These parameters are provided for each of Y, Cb and Cr separately and are listed in Table 39-9.

Component	Segment	Left	Right	Top	Bottom
Y	LUSEG	LULEFT	LURIGHT	LUTOP	LUBOTTOM
Cb	CBSEG	CBLEFT	CBRIGHT	CBTOP	CBBOTTOM
Cr	CRSEG	CRLEFT	CRRIGHT	CRTOP	CRBOTTOM

Table 39-9. Address Generation Registers for 2D Block Separated YUV Format

The 2D block addressing mode of the VIP is depending on the display array memory map of the Memory controller. Therefore, the locations of the segments should exist within the display array area. In addition, the Memory controller should be set to supports the display array area. For detailed information, see section ‘5.3.2 Display Array Area’.

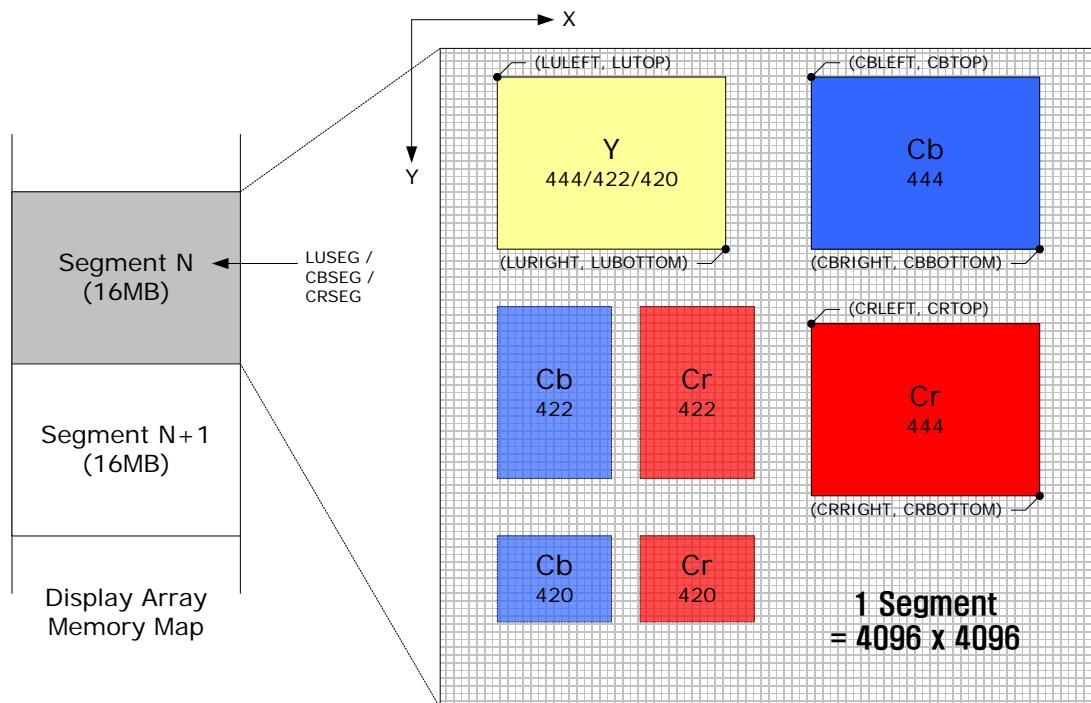


Figure 39-12. Address Generation for 2D Block Separated YUV Format

39.4.3 Flip & Rotation

Both the Clipper and the Decimator can horizontally/vertically flip or rotate output images. In Rotation, images are stored after they are rotated 90° clockwise. The *HFLIP* should be set as '1' for the Horizontal Flip, while the *VFLIP* should be set as '1' for the Vertical Flip. For a 90° Rotation, the *ROTATION* is set as '1'. Table 39-10 shows the resulting images depending on the *HFLIP*, the *VFLIP* and the *ROTATION*.

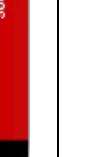
VFLIP		HFLIP		0	0	1	1
		0	0	0	1	0	1
0	0						
0	1						
1	0						
1	1						
ROTATION		0		1			

Table 39-10. Horizontal & Vertical Flip, Rotation

If the output format is the linear YUV 4:2:2 format, the Flip and Rotation functions are not supported.

39.4.4 Interlace Scan mode

If the *INTERLACENB* is set as '1' for interlace scan mode, the Clipper and Decimator store output images by field signals. The Clipper and Decimator can also select the polarity of a field signal by using *FIELDINV*. The Clipper supports frame-based output images, and outputs the images after automatically adjusting the start address and the start line depending on the field signal. For interlaced images, the Decimator only handles and stores even field data.

39.4.5 Pixels Alignment

The VIP Input and output image size should be aligned to 64 pixels.

39.5 Interrupt Generation

The VIP has three interrupt sources. The three interrupt sources are the HSINT, which generates an interrupt at the end of a horizontal sync, the VSINT, which generates an interrupt at the end of a vertical sync, and the ODINT, which generates an interrupt when the Clipper and Decimator operations are finished. The Pending bits and the Enable bits for each interrupt exist, and each register is listed in Table 39-11.

Interrupt	Enable bit	Pending bit	Condition
HSINT	HSINTENB	HSINTPEND	End of a horizontal sync pulse
VSINT	VSINTENB	VSINTPEND	End of a vertical sync pulse
ODINT	ODINTENB	ODINTPEND	Completion of the Clipper and Decimator operations

Table 39-11. Interrupt registers

39.6 Register Summary

BASE Address of VIP 0: 0xC0063000

BASE Address of VIP 1: 0xC0064000

Bit	R/W	Symbol	Description	Reset Value
VIP Configuration Register (VIP_CONFIG)				
<i>Address : VIP0 0xC0063000 / VIP1 0xC0064000</i>				
[15:10]	R	RESERVED	Reserved for future use.	6'b0
[9]	R/W	RESERVED	Reserved for future use. You have to write '0' only.	1'b0
[8]	R/W	EXTSYNCENB	Specifies the use of external sync signals. 0 : Embedded Sync 1 : External Sync	1'b0
[7:4]	R	RESERVED	Reserved for future use.	4'b0
[3:2]	R/W	DORDER	Specifies the order of input video data. 00 : Cb, Y0, Cr, Y1 10 : Y0, Cb, Y1, Cr 01 : Cr, Y1, Cb, Y0 11 : Y1, Cr, Y0, Cb	2'b0
[1]	R/W	DWIDTH	Specifies the bit-width of an input video signal 0: 16 bit 1: 8 bit	1'b0
[0]	R/W	VIPENB	VIP Enable 0 : Disable 1 : Enable	1'b0
VIP Interrupt Control Register (VIP_INTCTRL)				
<i>Address : VIP0 0xC0063002 / VIP1 0xC0064002</i>				
[15:10]	R	RESERVED	Reserved for future use.	6'b0
[9]	R/W	HSINTENB	Specifies the generation of an interrupt when an HSYNC event occurs. HSYNC events occur at the end of the HSYNC pulse. 0 : Disable 1 : Enable	1'b0
[8]	R/W	VSINTENB	Specifies the generation of an interrupt when a VSYNC event occurs. VSYNC events occur at the end of the VSYNC pulse. Therefore, the event occurs at every frame for Progressive input, and at every field for Interface input. 0 : Disable 1 : Enable	1'b0
[7:2]	R	RESERVED	Reserved for future use.	6'b0
[1]	R/W	HSINTPEND	Indicates the Pending status of the HSYNC interrupt. This bit always works regardless of the setting of the HSINTENB bit. Read> 0 : Not pending Write> 0 : No affect 1 : Pended 1 : Clear	1'b0
[0]	R/W	VSINTPEND	Indicates the Pending status of the VSYNC interrupt. This bit always works regardless of the setting of the VSINTENB bit. Read> 0 : Not pending Write> 0 : No affect 1 : Pended 1 : Clear	1'b0
VIP Interrupt Control Register (VIP_INTCTRL)				
<i>Address : VIP0 0xC0063004 / VIP1 0xC0064004</i>				
[15:10]	R	RESERVED	Reserved for future use.	6'b0
[9:8]	R/W	RESERVED	Reserved for future use. You have to write '3' only.	2'b0
[7:6]	R	RESERVED	Reserved for future use.	2'b0
[5]	R	LASTFIELD	Indicates the status of the internal Field signal that is updated at every FrameStart (the start of vertical active video). For the operation of this bit, the PCLKMODE is set as '1'. 0 : The last field is an odd field. 1 : The last field is an even field.	1'b0

Bit	R/W	Symbol	Description	Reset Value
[4]	R/W	DVALIDPOL	Selects the polarity of an external DVALID signal. This bit is valid only when the EXTDVENB is '1'. 0 : Active Low 1 : Active High	1'b0
[3]	R/W	EXTFIELDENB	Specifies the use of an external field signal. 0 : Disable 1 : Enable	1'b0
[2]	R/W	EXTDVENB	Specifies the use of an external DVALID signal. 0 : Disable 1 : Enable	1'b0
[1:0]	R/W	FIELDSEL	Selects a field signal. 00 : Bypass (Low is odd field) 01 : Invert (Low is even field) 10 : Fix 0 (odd field) 11 : Fix 1 (even field)	2'b0
VIP Sync Monitor Register (VIP_SYNCMON)				
<i>Address : VIP0 0xC0063006 / VIP1 0xC0064006</i>				
[15:2]	R	RESERVED	Reserved for future use.	14'b0
[1]	R	CURHSYNC	Indicates the status of the VSYNC created by the internal sync generator in Embedded Sync mode. This bit is valid only when the EXTSYNCENB is '0'. 0 : Inactivate 1 : Activate	1'b1
[0]	R	CURVSYNC	Indicates the status of the VSYNC created by the internal sync generator in Embedded Sync mode. This bit is valid only when the EXTSYNCENB is '0'. 0 : Inactivate 1 : Activate	1'b1
VIP Vertical Sync Start Register (VIP_VBEGIN)				
<i>Address : VIP0 0xC0063008 / VIP1 0xC0064008</i>				
[15:0]	R/W	VBEGIN	When the EXTSYNCENB is '1', this value is used for the creation of an internal vertical blank. This value specifies the number of lines in a section from the end of the vertical sync pulse to the end of the vertical blank. ▪ $VBEGIN = tVBP - 1$ When the EXTSYNCENB is '0', this value is used for the creation of an internal vertical sync pulse. This value specifies the number of lines in a section from the start of the vertical blank to the start of the vertical sync pulse. ▪ $VBEGIN = tVFP + 1$	16'b0
VIP Vertical Sync End Register (VIP_VEND)				
<i>Address : VIP0 0xC006300A / VIP1 0xC006400A</i>				
[15:0]	R/W	VEND	When the EXTSYNCENB is '1', this value is used for the creation of an internal vertical blank. This value specifies the number of lines in a section from the end of the vertical sync pulse to the start of the vertical blank. ▪ $VEND = tVBP + tAVH - 1$ When the EXTSYNCENB is '0', this value is used for the creation of an internal vertical sync pulse. This value specifies the number of lines in a section from the start of the vertical blank to the end of the vertical sync pulse. ▪ $VEND = tVFP + tVSW + 1$	16'b0
VIP Horizontal Sync Start Register (VIP_HBEGIN)				
<i>Address : VIP0 0xC006300C / VIP1 0xC006400C</i>				
[15:0]	R/W	HBEGIN	When the EXTSYNCENB is '1', this value is used for the creation of an internal horizontal blank. This value specifies the number of clocks in a section from the end of the horizontal sync pulse to the end of the horizontal blank. ▪ $HBEGIN = tHBP - 1$ When the EXTSYNCENB is '0', this value is used for the creation of an internal horizontal sync pulse. This value specifies the number of clocks in a section from the start of the horizontal blank to the start of the horizontal sync pulse.	16'b0

Bit	R/W	Symbol	Description	Reset Value
[7:2]	R	RESERVED	Reserved for future use.	6'b0
[1]	R/W	CLIPENB	Enables/disables the memory writing function of the Clipper block. This bit is valid only when the SEPENB is '1'. 0 : Disable 1 : Enable	1'b0
[0]	R/W	DECIENB	Enables/disables the memory writing function of the Decimator block. This bit is valid only when the SEPENB is '1'. 0 : Disable 1 : Enable	1'b0
VIP Operation Done Interrupt Register (VIP_ODINT)				
<i>Address : VIP0 0xC0063202 / VIP1 0xC0064202</i>				
[15:9]	R	RESERVED	Reserved for future use.	7'b0
[8]	R/W	ODINTENB	Specifies the generation of an interrupt when the Clipper/Decimator complete a frame/field. 0 : Disable 1 : Enable	1'b0
[7:1]	R	RESERVED	Reserved for future use.	7'b0
[0]	R/W	ODINTPEND	Indicates the Pending status of Clipper & Decimator Operation Done events. This bit always operates regardless of the setting of the ODINTENB bit. Read> 0 : Not pended 1 : Pended Write> 0 : No affect 1 : Clear	1'b0
VIP Image Width Register (VIP_IMGWIDTH)				
<i>Address : VIP0 0xC0063204 / VIP1 0xC0064204</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	IMGWIDTH	Specifies the width of input images in pixel units. When EXSYNCENB is '0', you have to set it as 'image width + 2'.	12'b0
VIP Image Height Register (VIP_IMGHEIGHT)				
<i>Address : VIP0 0xC0063206 / VIP1 0xC0064206</i>				
[15:11]	R	RESERVED	Reserved for future use.	5'b0
[10:0]	R/W	IMGHEIGHT	Specifies the height of input images in line units.	11'b0
VIP Clipper Left Register (CLIP_LEFT)				
<i>Address : VIP0 0xC0063208 / VIP1 0xC0064208</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CLIPLEFT	Specifies the X-coordinate on the top left corner of the area to be clipped, in pixels. The clipping width (CLIPRIGHT - CLIPLEFT) must be a multiple of 16.	12'b0
VIP Clipper Left Register (CLIP_RIGHT)				
<i>Address : VIP0 0xC006320A / VIP1 0xC006420A</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CLIPRIGHT	Specifies the X-coordinate on the bottom right corner of the area to be clipped, in pixels. It should have a greater value than the CLIPLEFT . The clipping width (CLIPRIGHT - CLIPLEFT) must be a multiple of 16.	12'b0
VIP Clipper Top Register (CLIP_TOP)				
<i>Address : VIP0 0xC006320C / VIP1 0xC006420C</i>				
[15:11]	R	RESERVED	Reserved for future use.	5'b0
[10:0]	R/W	CLITOP	Specifies the Y-coordinate on the top left corner of the area to be clipped, in pixels. The clipping height (CLIPBOTTOM - CLITOP) must be an even number.	11'b0
VIP Clipper Bottom Register (CLIP_BOTTOM)				
<i>Address : VIP0 0xC006320E / VIP1 0xC006420E</i>				
[15:11]	R	RESERVED	Reserved for future use.	5'b0

Bit	R/W	Symbol	Description	Reset Value
[10:0]	R/W	CLIPBOTTOM	Specifies the Y-coordinate on the bottom right corner of the area to be clipped, in pixels. It should have a greater value than the CLIPTOP . The clipping height (CLIPBOTTOM – CLIPTOP) must be an even number.	11'b0
VIP Decimator Target Width Register (DECI_TARGETW)				
<i>Address : VIP0 0xC0063210 / VIP1 0xC0064210</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	TARGETW	Specifies the width of the Decimator output image, in pixels. The width of the output image should be narrower than that of the clipped input image and be a multiple of 16.	12'b0
VIP Decimator Target Height Register (DECI_TARGETH)				
<i>Address : VIP0 0xC0063212 / VIP1 0xC0064212</i>				
[15:11]	R	RESERVED	Reserved for future use.	5'b0
[10:0]	R/W	TARGETH	Specifies the height of the Decimator output image, in lines. The height of the output image should be lower than that of the clipped input image and be an even number.	11'b0
VIP Decimator Delta Width Register (DECI_DELTAW)				
<i>Address : VIP0 0xC0063214 / VIP1 0xC0064214</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	DELTAW	Specifies the width difference between the input image and the output image of the Decimator in pixel units. • DELTAW = (CLIPRIGHT – CLILEFT) - TARGETW	12'b0
VIP Decimator Delta Height Register (DECI_DELTAH)				
<i>Address : VIP0 0xC0063216 / VIP1 0xC0064216</i>				
[15:11]	R	RESERVED	Reserved for future use.	5'b0
[10:0]	R/W	DELTAH	Specifies the line difference between the input image and the output image of the Decimator in line units. • DELTAH = (CLIPBOTTOM – CLIPTOP) - TARGETH	11'b0
VIP Decimator Clear Width Register (DECI_CLEARW)				
<i>Address : VIP0 0xC0063218 / VIP1 0xC0064218</i>				
[15:11]	R	RESERVED	Reserved for future use.	3'b0
[12:0]	R/W	CLEARW	Specifies the difference between the width of the Decimator output image and the DELTAW in pixel units. This value has 2's complement format. • CLEARW = TARGETW - DELTAW	13'b0
VIP Decimator Clear Height Register (DECI_CLEARH)				
<i>Address : VIP0 0xC006321A / VIP1 0xC006421A</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CLEARH	Specifies the difference between the height of the Decimator output image and the DELTAH in line units. This value has 2's complement format. • CLEARH = TARGETH - DELTAH	12'b0
VIP Decimator Lu Segment Register (DECI_LUSEG)				
<i>Address : VIP0 0xC006321C / VIP1 0xC006421C</i>				
[15:8]	R	RESERVED	Reserved for future use.	8'b0
[7:0]	R/W	LUSEG	Specifies the index of the segment where the output Y data of the Decimator is stored. This value has a Linear address [31:24].	8'b0
VIP Decimator Cr Segment Register (DECI_CRSEG)				
<i>Address : VIP0 0xC006321E / VIP1 0xC006421E</i>				
[15:8]	R	RESERVED	Reserved for future use.	8'b0
[7:0]	R/W	CRSEG	Specifies the index of the segment where the output Cr data of the Decimator is stored.	8'b0

Bit	R/W	Symbol	Description	Reset Value
			This value has a Linear address [31:24].	
VIP Decimator Cb Segment Register (DECI_CBSEG)				
<i>Address : VIP0 0xC0063220 / VIP1 0xC0064220</i>				
[15:8]	R	RESERVED	Reserved for future use.	8'b0
[7:0]	R/W	CBSEG	Specifies the index of the segment where the output Cb data of the Decimator is stored. This value has a Linear address [31:24].	8'b0
VIP Decimator Format Register (DECI_FORMAT)				
<i>Address : VIP0 0xC0063222 / VIP1 0xC0064222</i>				
[15:2]	R	RESERVED	Reserved for future use.	14'b0
[1:0]	R/W	FORMATSEL	Specifies the output format of the Decimator. 00 : 2D block separated YUV 4:2:0 01 : 2D block separated YUV 4:2:2 10 : 2D block separated YUV 4:4:4 11 : Reserved	2'b0
VIP Decimator Rotation & Flip Register (DECI_ROTFLIP)				
<i>Address : VIP0 0xC0063224 / VIP1 0xC0064224</i>				
[15:3]	R	RESERVED	Reserved for future use.	13'b0
[2]	R/W	ROTATION	Specifies whether to rotate the output image of Decimator 90 degree clockwise. 0 : Disable 1 : Enable	1'b0
[1]	R/W	VFLIP	Specifies whether to flip the output image of Decimator vertically. 0 : Disable 1 : Enable	1'b0
[0]	R/W	HFLIP	Specifies whether to flip the output image of Decimator horizontally. 0 : Disable 1 : Enable	1'b0
VIP Decimator Lu Left Register (DECI_LULEFT)				
<i>Address : VIP0 0xC0063226 / VIP1 0xC0064226</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	LULEFT	Specifies the X-coordinate on the top left corner of the area in the segment where the output Y data of the Decimator is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cr Left Register (DECI_CRLEFT)				
<i>Address : VIP0 0xC0063228 / VIP1 0xC0064228</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CRLEFT	Specifies the X-coordinate on the top left corner of the area in the segment where the output Cr data of the Decimator is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cb Left Register (DECI_CBLEFT)				
<i>Address : VIP0 0xC006322A / VIP1 0xC006422A</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CBLEFT	Specifies the X-coordinate on the top left corner of the area in the segment where the output Cb data of the Decimator is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Decimator Lu Right Register (DECI_LURIGHT)				
<i>Address : VIP0 0xC006322C / VIP1 0xC006422C</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	LURIGHT	Specifies the X-coordinate on the bottom right corner of the area in the segment where the output Y data of the Decimator is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cr Right Register (DECI_CRRIGHT)				
<i>Address : VIP0 0xC006322E / VIP1 0xC006422E</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0

Bit	R/W	Symbol	Description	Reset Value
[11:0]	R/W	CRRIGHT	Specifies the X-coordinate on the bottom right corner of the area in the segment where the output Cr data of the Decimator is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cb Right Register (DECI_CBRIGHT)				
<i>Address : VIP0 0xC0063230 / VIP1 0xC0064230</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CBRIGHT	Specifies the X-coordinate on the bottom right corner of the area in the segment where the output Cb data of the Decimator is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Decimator Lu Top Register (DECI_LUTOP)				
<i>Address : VIP0 0xC0063232 / VIP1 0xC0064232</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	LUTOP	Specifies the Y-coordinate on the top left corner of the area in the segment where the output Y data of the Decimator is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cr Top Register (DECI_CRTOP)				
<i>Address : VIP0 0xC0063234 / VIP1 0xC0064234</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CRTOP	Specifies the Y-coordinate on the top left corner of the area in the segment where the output Cr data of the Decimator is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cb Top Register (DECI_CBTOP)				
<i>Address : VIP0 0xC0063236 / VIP1 0xC0064236</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CBTOP	Specifies the Y-coordinate on the top left corner of the area in the segment where the output Cb data of the Decimator is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Decimator Lu Bottom Register (DECI_LUBOTTOM)				
<i>Address : VIP0 0xC0063238 / VIP1 0xC0064238</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	LUBOTTOM	Specifies the Y-coordinate on the bottom right corner of the area in the segment where the output Y data of the Decimator is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cr Bottom Register (DECI_CRBOTTOM)				
<i>Address : VIP0 0xC006323A / VIP1 0xC006423A</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CRBOTTOM	Specifies the Y-coordinate on the bottom right corner of the area in the segment where the output Cr data of the Decimator is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Decimator Cb Bottom Register (DECI_CBBOTTOM)				
<i>Address : VIP0 0xC006323C / VIP1 0xC006423C</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CBBOTTOM	Specifies the Y-coordinate on the bottom right corner of the area in the segment where the output Cb data of the Decimator is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Clipper Lu Segment Register (CLIP_LUSEG)				
<i>Address : VIP0 0xC006323E / VIP1 0xC006423E</i>				
[15:8]	R	RESERVED	Reserved for future use.	8'b0
[7:0]	R/W	LUSEG	Specifies the index of the segment where the output Y data of the Clipper is stored. This value has a Linear address [31:24].	8'b0
VIP Clipper Cr Segment Register (CLIP_CRSEG)				
<i>Address : VIP0 0xC0063240 / VIP1 0xC0064240</i>				

Bit	R/W	Symbol	Description	Reset Value
[15:8]	R	RESERVED	Reserved for future use.	8b0
[7:0]	R/W	CRSEG	Specifies the index of the segment where the output Cr data of the Clipper is stored. This value has a Linear address [31:24].	8b0
VIP Clipper Cb Segment Register (CLIP_CBSEG)				
<i>Address : VIP0 0xC0063242 / VIP1 0xC0064242</i>				
[15:8]	R	RESERVED	Reserved for future use.	8b0
[7:0]	R/W	CBSEG	Specifies the index of the segment where the output Cb data of the Clipper is stored. This value has a Linear address [31:24].	8b0
VIP Clipper Format Register (CLIP_FORMAT)				
<i>Address : VIP0 0xC0063244 / VIP1 0xC0064244</i>				
[15:2]	R	RESERVED	Reserved for future use.	14b0
[1:0]	R/W	FORMATSEL	Specifies the output format of the Clipper. 00 : 2D block separated YUV 4:2:0 01 : 2D block separated YUV 4:2:2 10 : 2D block separated YUV 4:4:4 11 : Reserved	2b0
VIP Clipper Rotation & Flip Register (CLIP_ROTFLIP)				
<i>Address : VIP0 0xC0063246 / VIP1 0xC0064246</i>				
[15:3]	R	RESERVED	Reserved for future use.	13b0
[2]	R/W	ROTATION	Specifies whether to rotate the output image of Clipper 90 degree clockwise. When YUYVNB is '0', you have to disable the rotation function. 0 : Disable 1 : Enable	1b0
[1]	R/W	VFLIP	Specifies whether to flip the output image of Clipper vertically. 0 : Disable 1 : Enable	1b0
[0]	R/W	HFLIP	Specifies whether to flip the output image of Clipper horizontally. 0 : Disable 1 : Enable	1b0
VIP Clipper Lu Left Register (CLIP_LULEFT)				
<i>Address : VIP0 0xC0063248 / VIP1 0xC0064248</i>				
[15:12]	R	RESERVED	Reserved for future use.	4b0
[11:0]	R/W	LULEFT	Specifies the X-coordinate on the top left corner of the area in the segment where the output Y data of the Clipper is stored. The X-coordinate should be a multiple of 8.	12b0
VIP Clipper Cr Left Register (CLIP_CRLEFT)				
<i>Address : VIP0 0xC006324A / VIP1 0xC006424A</i>				
[15:12]	R	RESERVED	Reserved for future use.	4b0
[11:0]	R/W	CRLEFT	Specifies the X-coordinate on the top left corner of the area in the segment where the output Cr data of the Clipper is stored. The X-coordinate should be a multiple of 8.	12b0
VIP Clipper Cb Left Register (CLIP_CBLEFT)				
<i>Address : VIP0 0xC006324C / VIP1 0xC006424C</i>				
[15:12]	R	RESERVED	Reserved for future use.	4b0
[11:0]	R/W	CBLEFT	Specifies the X-coordinate on the top left corner of the area in the segment where the output Cb data of the Clipper is stored. The X-coordinate should be a multiple of 8.	12b0
VIP Clipper Lu Right Register (CLIP_LURIGHT)				
<i>Address : VIP0 0xC006324E / VIP1 0xC006424E</i>				
[15:12]	R	RESERVED	Reserved for future use.	4b0
[11:0]	R/W	LURIGHT	Specifies the X-coordinate on the bottom right corner of the area in the segment where the output Y data of the Clipper is stored. The X-coordinate should be a multiple of 8.	12b0

Bit	R/W	Symbol	Description	Reset Value
VIP Clipper Cr Right Register (CLIP_CRRIGHT)				
<i>Address : VIP0 0xC0063250 / VIP1 0xC0064250</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CRRIGHT	Specifies the X-coordinate on the bottom right corner of the area in the segment where the output Cr data of the Clipper is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Clipper Cb Right Register (CLIP_CBRIGHT)				
<i>Address : VIP0 0xC0063252 / VIP1 0xC0064252</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CBRIGHT	Specifies the X-coordinate on the bottom right corner of the area in the segment where the output Cb data of the Clipper is stored. The X-coordinate should be a multiple of 8.	12'b0
VIP Clipper Lu Top Register (CLIP_LUTOP)				
<i>Address : VIP0 0xC0063254 / VIP1 0xC0064254</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	LUTOP	Specifies the Y-coordinate on the top left corner of the area in the segment where the output Y data of the Clipper is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Clipper Cr Top Register (CLIP_CRTOP)				
<i>Address : VIP0 0xC0063256 / VIP1 0xC0064256</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CRTOP	Specifies the Y-coordinate on the top left corner of the area in the segment where the output Cr data of the Clipper is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Clipper Cb Top Register (CLIP_CBTOP)				
<i>Address : VIP0 0xC0063258 / VIP1 0xC0064258</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CBTOP	Specifies the Y-coordinate on the top left corner of the area in the segment where the output Cb data of the Clipper is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Clipper Lu Bottom Register (CLIP_LUBOTTOM)				
<i>Address : VIP0 0xC006325A / VIP1 0xC006425A</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	LUBOTTOM	Specifies the Y-coordinate on the bottom right corner of the area in the segment where the output Y data of the Clipper is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Clipper Cr Bottom Register (CLIP_CRBOTTOM)				
<i>Address : VIP0 0xC006325C / VIP1 0xC006425C</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CRBOTTOM	Specifies the Y-coordinate on the bottom right corner of the area in the segment where the output Cr data of the Clipper is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Clipper Cb Bottom Register (CLIP_CBBOTTOM)				
<i>Address : VIP0 0xC006325E / VIP1 0xC006425E</i>				
[15:12]	R	RESERVED	Reserved for future use.	4'b0
[11:0]	R/W	CBBOTTOM	Specifies the Y-coordinate on the bottom right corner of the area in the segment where the output Cb data of the Clipper is stored. The Y-coordinate should be a multiple of 8.	12'b0
VIP Scan Mode Register (VIP_SCANMODE)				
<i>Address : VIP0 0xC0063260 / VIP1 0xC0064260</i>				
[15:2]	R	RESERVED	Reserved for future use.	14'b0
[1]	R/W	INTERLACEENB	Specifies the scan mode of an input image.	1'b0

Bit	R/W	Symbol	Description	Reset Value
			0 : Progressive scan mode 1 : Interlace scan mode	
[0]	R/W	FIELDINV	Specifies the polarity of the field signal transmitted from the VIP block to the Clipper and to the Decimator. 0 : Bypass (Low is odd field) 1 : Invert (Low is even field)	1'b0
VIP Clipper Linear YUV Enable Register (CLIP_YUYVENB)				
<i>Address : VIP0 0xC0063262 / VIP1 0xC0064262</i>				
[15:1]	R	RESERVED	Reserved for future use.	15'b0
[0]	R/W	YUYVENB	Specifies the output format of the Clipper as the linear format or the 2D block separated format. 0 : 2D Block Separated YUV format 1 : Linear YUV(YUYV) format	1'b0
VIP Clipper Linear Base Address High Register (CLIP_BASEADDRH)				
<i>Address : VIP0 0xC0063264 / VIP1 0xC0064264</i>				
[15:0]	R/W	BASEADDRH	Specifies the upper [31:16] of the base address where the linear YUV data of the Clipper is stored. In interface scan mode, the address to store an even field image is calculated automatically. The value is used only when the YUYVENB is '1'.	16'b0
VIP Clipper Linear Base Address Low Register (CLIP_BASEADDRL)				
<i>Address : VIP0 0xC0063266 / VIP1 0xC0064266</i>				
[15:0]	R/W	BASEADDRL	Specifies the lower [15:0] of the base address where the linear YUV data of the Clipper is stored. This value must be a multiple of 8. In interface scan mode, the address to store an even field image is calculated automatically. The value is used only when the YUYVENB is '1'.	16'b0
VIP Clipper Linear Stride High Register (CLIP_STRIDEH)				
<i>Address : VIP0 0xC0063268 / VIP1 0xC0064268</i>				
[15:0]	R/W	STRIDEH	Specifies the upper [31:16] of the stride where the linear YUV data of the Clipper is stored. In general, the stride has the byte size of a line. Since this value is automatically doubled in interlace scan mode, the value in progressive scan mode should be specified. The value is used only when the YUYVENB is '1'.	16'b0
VIP Clipper Linear Stride Low Register (CLIP_STRIDEL)				
<i>Address : VIP0 0xC006326A / VIP1 0xC006426A</i>				
[15:0]	R/W	STRIDEL	Specifies the lower [15:0] of the stride where the linear YUV data of the Clipper is stored. In general, the stride has the byte size of a line. Since this value is automatically doubled in interlace scan mode, the value in progressive scan mode should be specified. The value is used only when the YUYVENB is '1'.	16'b0

Section 40. Multi-Format Video Codec

40.1 Overview

40.1.1 Features

The multi format video codec (hereinafter referred to as "VPU") is a full HD multi-standard video IP for consumer multimedia products such as HDTVs, HD set-top boxes, and HD DVD players. It can decode compressed video in a format of H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-1/2, MPEG-4 SP/ASP, H.263P3, DivX/XviD, VP8, Theora, AVS, RV-8/9/10, and JPEG (max. 8192x8192) It can also perform H.264, MPEG-4, and H.263 encoding up to Full-HD 1920x1088 (max. 8192x8192 JPEG) resolution. The VPU can perform simultaneous multiple realtime encoding, decoding, or both encoding and decoding of different format video streams at multiple resolutions.

The VPU contains a 16-bit DSP called BIT processor. The BIT processor communicates with a host CPU through a host interface and controls the other sub-blocks of the VPU. The host CPU requires low resources under 1 MIPS, because all of the functions such as bitstream parsing, video hardware sub-blocks control and error resilience are implemented in the BIT processor. Moreover it is designed to optimally share most of the sub-blocks that are used in common for video processing, which contributes to the ultra low power and low gate count.

It is connected with a host CPU system via 32-bit AMBA 3 APB bus for system control and 128-bit AMBA3 AXI for data. There are two 128-bit AXI buses: primary and secondary. The secondary bus can be connected to on-chip memories to achieve high performance.

40.2 Functional Description

40.2.1 List of video codecs

The following table shows many different video standards supported by VPU.

	Standard	Profile	Level	Max. Resolution	Min. Resolution	Bitrate
Encoder	H.264	Baseline	4.0	1920x1088	96x16	20Mbps
	MPEG-4	SP	5/6	1920x1088	96x16	20Mbps
	H.263	Profile3	70	1920x1088	96x16	20Mbps
Decoder	H.264	BP/MP/HP	4.2	1920x1088	16x16	50Mbps
	MPEG-4	ASP		1920x1088	16x16	40Mbps
	H.263	Profile3		1920x1088	16x16	20Mbps
	VC-1	SP/MP/AP	3	1920x1088	16x16	45Mbps
	MPEG-1/2	MP	High	1920x1088	16x16	80Mbps
	DivX/XviD	Home theater		1920x1088	16x16	40Mbps
	VP8			1920x1088	16x16	20Mbps
	Theora			1280x720	16x16	20Mbps
	AVS	Jizhun	6.2	1920x1088	16x16	40Mbps
	RV	8/9/10		1920x1088	16x16	40Mbps
Encoder	MJPEG	Baseline		8192x8192	16x16	160Mpel/s at YUV422
Decoder	MJPEG	Baseline		1920x1088	16x16	120Mpel/s at YUV444

Table 40-1. Supported Video Standards

40.2.2 Supported video encoding tools

H.264/AVC BP/CBP Encoder

- Compatible with the ITU-T Recommendation H.264 specification
- The encoder uses only one reference frame for the motion estimation.
- 1/4-pel accuracy motion estimation with programmable search range up to [+/-128, +/-64]
- Search range is reconfigurable by SW
 - Horizontal(-128 ~ 127), Vertical(-64 ~ 63)
 - Horizontal(-64 ~ 63), Vertical(-32 ~ 31)
 - Horizontal(-32 ~ 31), Vertical(-16 ~ 15)
 - Horizontal(-16 ~ 15), Vertical(-16 ~ 15)
- 16x16, 16x8, 8x16 and 8x8 block sizes are supported.
- Available block sizes can be configurable.
- Intra-prediction

- Luma I4x4 Mode : 9 modes
- Luma I16x16 Mode : 3 modes (Vertical, Horizon, DC)
- Chroma Mode : 3 modes (Vertical, Horizon, DC)
- Minimum encoding image size is 96 pixels in horizontal and 16 pixels in vertical.
- The encoder supports the following error resilience tools: video packet (fixed number of bits, and fixed number of macroblocks), CIR (Cyclic Intra Refresh), and multi-slice structure.
- FMO/ASO tool of H.264 is not supported.
- The encoder rate control is configurable for low-delay and long-delay, and configurable from macroblock-level rate control to frame-level rate control.
- Field encoding is available without PAFF, MBAFF.

MPEG4-SP Encoder

- Compatible with the ISO/IEC 14496-2 specification
- MV with unrestricted motion vector
- AC/DC prediction
- 1/2-pel accuracy motion estimation with search range up to [+/-128, +/-64]
- Search range is reconfigurable by SW
 - Horizontal(-128 ~ 127), Vertical(-64 ~ 63)
 - Horizontal(-64 ~ 63), Vertical(-32 ~ 31)
 - Horizontal(-32 ~ 31), Vertical(-16 ~ 15)
 - Horizontal(-16 ~ 15), Vertical(-16 ~ 15)
- Error resilience tools such as re-sync marker, data-partitioning with reversible VLC.

H.263 P0/P3 (Interactive and Streaming Wireless Profile) Encoder

- MV with unrestricted motion vector mode compliant to Annex D
- Search range is -16 ~ 15 in horizontal and -16 ~ 15 in vertical
- H.263 Baseline profile + Annex J, K (RS=0 and ASO=0), and T

40.2.3 Supported video decoding tools

H.264/AVC Decoder

- Fully compatible with the ITU-T Recommendation H.264 specification in BP, MP and HP.
- Supports MVC Stereo High profile
- Supports CABAC/CAVLC
- Variable block size (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4)
- Error detection, concealment and error resilience tools with FMO/ASO support

VC-1/WMV-9 Decoder

- Supports all VC-1 profile features - SMPTE Proposed SMPTE Standard for Television: VC-1 Compressed

- Video Bitstream format and Decoding Process
- Supports Simple/Main/Advanced Profile
- Supports multi-resolution (Dynamic resolution) without scaling that returns related information

MPEG-4 Decoder

- Fully compatible with the ISO/IEC 14496-2 specification in SP/ASP except GMC(Global motion compensation)
- Full XviD compatibility
- Support for short video header

DivX Decoder

- Fully DivX certifiable for the Handheld/Portable/Home Theater/High Definition profiles for both progressive and interlaced format

Sorenson Spark Decoder

- Fully compatible with Sorenson Spark decoder specification

H.263 V2 (Interactive and Streaming Wireless Profile, Profile 3) Decoder

- H.263 Baseline profile + Annex I, J, K (except RS/ASO), and T

MPEG-1/MPEG-2

- Fully compatible with ISO/IEC 13818-2 MPEG2 specification in Main Profile
- Support I,P and B frame
- Support field coded picture (interlaced) and fame coded picture

AVS Decoder

- Supports Jizhun profile level 6.2 (exclude 422 case)

Real Video 10 Decoder

- Fully compatible with RV-8/9/10 except re-sampling feature
- Minimum decoding size is 32x32 pixels.

VP8 Decoder

- Fully compatible with VP8 decoder specification
- Supporting both simple and normal in-loop deblocking

Theora Decoder

- Fully compatible with Theora decoder specification

40.2.4 Supported JPEG tools

MJPEG Baseline Process Encoder and Decoder

- Baseline ISO/IEC 10918-1 JPEG compliance
- Support 1 or 3 color components
- 3 component in a scan (interleaved only)
- 8 bit samples for each component
- Support 4:2:0, 4:2:2, 2:2:4, 4:4:4 and 4:0:0 color format (max. six 8x8 blocks in one MCU)
- Minimum encoding size is 16x16 pixels.

40.2.5 Non-codec related features

Value added features

- De-ringing(MPEG-2/4 only), rotator/mirroring
- Built-in de-blocking filter for MPEG-2/MPEG-4 and DivX
- Pre/Post rotator/mirror

Programmability

- The VPU embeds 16-bit DSP processor dedicated to processing bitstream and controlling their video hardwares.
- General purpose registers and interrupt for communication between a host processor and the video IP

Optimal external memory accesses

- Configurable frame buffer formats (linear or tiled) for longer burst-length
- 2D cache for motion estimation and compensation to reduce external memory accesses
- Secondary AXI port for on-chip memory to enhance performance

Section 41. 3D Graphic Engine

41.1 Overview

The 3D graphic engine of NXP4330D/Q is Mali-400 MP Core GPU. The Mali-400 MP Core GPU is a hardware accelerator for 2D and 3D graphics systems.

The GPU consist of

- Two Pixel Processors (PPs)
- a Geometry Processor (GP)
- a 32Kbyte Level 2 Cache (L2)
- a Memory Management Unit (MMU) for each GP and PP
- a Power Management Unit (PMU).

The GPU and its associated software is compatible with the following graphics standards:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1.

41.1.1 Pixel processor features

The pixel processor features are:

- each pixel processor used processes a different tile, enabling a faster turnaround
- programmable fragment shader
- alpha blending
- complete non-power-of-2 texture support
- cube mapping
- fast dynamic branching
- fast trigonometric functions, including arctangent
- full floating-point arithmetic
- framebuffer blend with destination Alpha
- indexable texture samplers
- line, quad, triangle and point sprites
- no limit on program length
- perspective correct texturing
- point sampling, bilinear, and trilinear filtering
- programmable mipmap level-of-detail biasing and replacement
- stencil buffering, 8-bit
- two-sided stencil
- unlimited dependent texture reads
- 4-level hierarchical Z and stencil operations
- Up to 512 times Full Scene Anti-Aliasing (FSAA). 4x multisampling times 128x supersampling

- 4-bit per texel compressed texture format

41.1.2 Geometry processor features

The geometry processor features are:

- programmable vertex shader
- flexible input and output formats
- autonomous operation tile list generation
- indexed and non-indexed geometry input
- primitive constructions with points, lines, triangles and quads.

41.1.3 Level 2 cache controller features

The L2 cache controller features are:

- 32KB 4-way set-associative
- supports up to 32 outstanding AXI transactions
- implements a standard pseudo-LRU algorithm
- cache line and line fill burst size is 64 bytes
- supports eight to 64bytes uncached read bursts and write bursts
- 128-bit interface to memory sub-system
- support for hit-under-miss and miss-under-miss with the only limitation of AXI ordering rules

41.1.4 MMU

The MMU features are:

- accesses control registers through the bus infrastructure to configure the memory system
- each processor has its own MMU to control and translate memory accesses that the GPU initiates

41.1.5 PMU

The PMU features are:

- programmable power management
- powers up and down each GP, PP and Level 2 cache controller separately
- controls the clock, isolation and power of each device
- provides an interrupt when all requested devices are powered up

41.2 Operation

41.2.1 Clock

The NXP4330D/Q has a clock for Mali-400 MP Core. The operation frequency can be up to 333 MHz. See the system controller and clock controller for setting up the Mali-400 MP Core clock.

41.2.2 Reset

The NXP4330D/Q has a reset for Mali-400 MP Core. See the reset controller for setting up the Mali-400 MP Core reset.

41.2.3 Interrupt

The NXP4330D/Q has an interrupt number for Mali-400 MP Core. See the interrupt controller for setting up the Mali-400 MP Core reset.

Section 42. Crypto Engine

42.1 Overview

Crypto Engine block executes AES, DES, HASH Encryption and Decryption.

42.1.1 Features

- Big Endian Encryption & Decryption
- Supports DMA Interface
- Supports AES ECB, CBC, CTR -128,192,256 Mode
- Supports DES ECB, CBC -64 Mode
- Supports 3DES -64 Mode
- Supports HASH Mode (SHA1, MD5)
- Supports input share Mode (AES & HASH)
- Supports AES and HASH working at the same time (Refer Figure 42-2)

42.1.2 Block Diagram

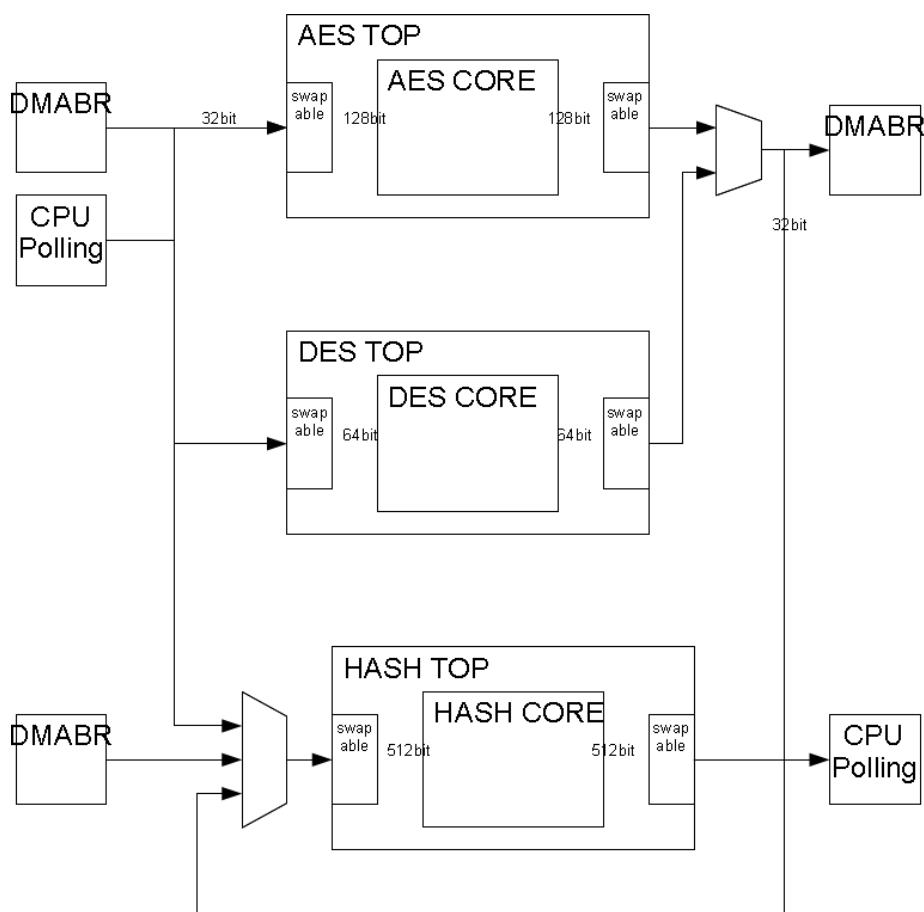


Figure 42-1. CRYPTO Block Diagram

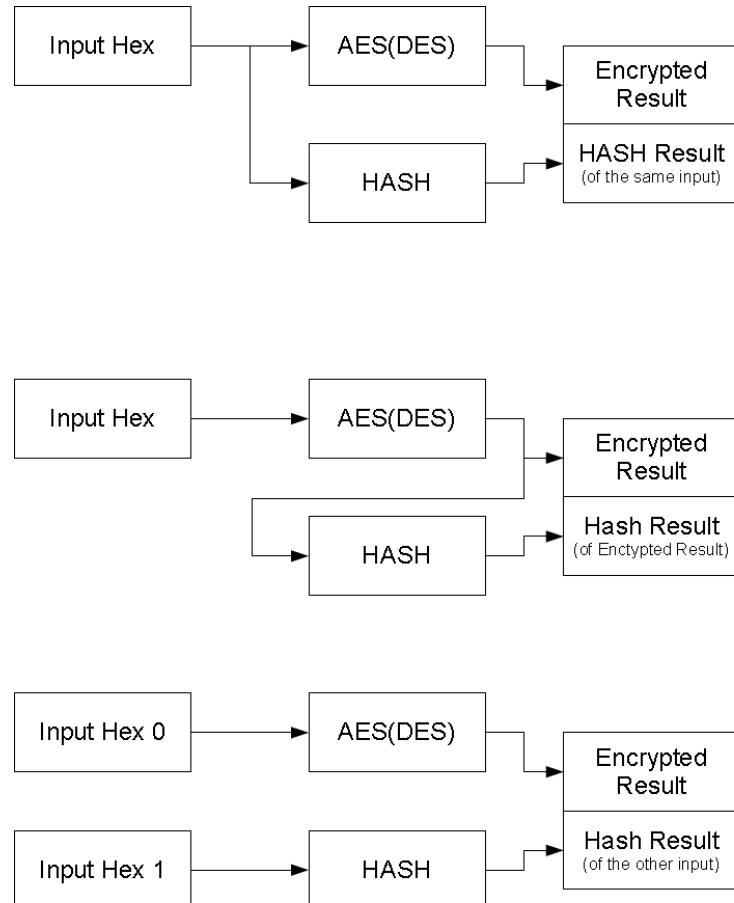


Figure 42-2. CRYPTO AES & HASH Operation Scenario

42.2 Functional Description

Polling Mode

In polling mode, CPU can write/read the register directly.

Access Sequence:

- 1) Set AES Key
- 2) Set Initial Vector and Set CPU_AES_LOADCNT with 1
- 3) Set AES_TESTIN
- 4) Set AES Control Register & Enable
- 5) Set CPU_AES_LOADCNT with 0
- 6) Set IDLC_ANNY WITH 1 (**AES_START**)
- 7) WAIT FOR IDLE_AES=1
- 8) Get Result Vector

Mode

In Channel DMA mode, users need to DMA Mode Enable.

42.3 Register Summary

Bit	R/W	Symbol	Description	Reset Value
CRYPTO Control register (CRT_CTRL0)				
Address :C001 5000h				
[31:11]	R	RESERVED	Reserved	21'b0
[10]	R/W	CPU_INT_ENB	Enable the Interrupt 0: Disable 1: Enable	1b0
[9]	R/W	CPU_INT_MASK	Masking the Interrupt 0: Masked 1: Not Masked	1b0
[8]	R/W	CPU_DMAW_SRC	Specifies the DMA En/Decryption Mode 0: AES 1: DES	1b0
[7]	R	IDLE_HASHCORE	Indicates the HASH CORE is Idle 0: Not Idle 1: Idle	
[6]	R	RESERVED	Reserved	1b0
[5]	R/W	INTPEND	Read : Interrupt Pending Bit Write 1 : Interrupt Pending Clear	
[4]	W	CPU_DES_LOADCNT	Users must this bit to 1 after users set the DES Initial Value.	
[3]	W	CPU_AES_LOADCNT	Users must this bit to 1 after users set the AES Initial Value.	
[2]	R/W	IDLE_HASH	Indicates the HASH is Idle Write 1 : HASH Start	
[1]	R/W	IDLE_DES	Indicates the DES is Idle Write 1 : DES Start	
[0]	R/W	IDLE_AES	Indicates the AES is Idle Write 1 : AES Start	
CRYPTO AES Control register (AES_CTRL0)				
Address :C001 500Ch				
[31:16]	R	RESERVED	Reserved	16'b0
[15]	R/W	AES_SELKEY	Select the AES KEY Mode 0: CPU configuration 1: ECID AESKEY	1b0
[14:10]	R	RESERVED	Reserved	5'b0
[9]	R/W	AES_SWAPOUT	Enable the AES Output Swap 0: Not Swap 1: Masked	1b0
[8]	R/W	AES_SWAPIN	Enable the AES Input Swap 0: Not Swap 1: Masked	1b0
[7:6]	R/W	AES_BLKMODE	Specifies the AES Block Mode 0: ECB 1: CBC 2: CTR 3: Reserved	2b0
[5:4]	R/W	AES_MODE	Specifies the AES bit Mode 0: 128 bit 1: 192 bit 2: 256 bit 3: Reserved	2b0
[3]	R/W	AES_128CNT	Enable the AES 128 bit Counter	1b0
[2]	R/W	AES_DMAMODE	Enable the AES DMA Interface 0: Disable 1: Enable	1b0

Bit	R/W	Symbol	Description	Reset Value
[1]	R/W	AES_ENC	Specifies the AES Encoding Mode (Encryption/Decryption) 0 : Decryption 1 : Modulation	1'b0
[0]	R/W	AES_ENB	Enable the AES Mode 0 : Disable 1 : Enable	1'b0
CRYPTO AES INIT vector register 0 (AES_iv0)				
Address :C001 5010h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_IV[127:96]. AES Initial vector	32'b0
CRYPTO AES INIT vector register 1 (AES_iv1)				
Address :C001 5014h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_IV[95:64]. AES Initial vector	32'b0
CRYPTO AES INIT vector register 2 (AES_iv2)				
Address :C001 5018h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_IV[63:32]. AES Initial vector	32'b0
CRYPTO AES INIT vector register 3 (AES_iv3)				
Address :C001 501Ch				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_IV[31:0]. AES Initial vector	32'b0
CRYPTO AES key register 0 (AES_key0)				
Address :C001 5030h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[255:224]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES key register 1 (AES_key1)				
Address :C001 5034h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[223:192]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES key register 2 (AES_key2)				
Address :C001 5038h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[192:160]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES key register 3 (AES_key3)				
Address :C001 503Ch				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[159:128]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES key register 4 (AES_key4)				
Address :C001 5040h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[127:96]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES key register 5 (AES_key5)				
Address :C001 5044h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[95:64]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES key register 6 (AES_key6)				
Address :C001 5048h				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[63:32]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES key register 7 (AES_key7)				
Address :C001 504Ch				
[31:0]	R/W	O_CPU_AES_IV	O_CPU_AES_key[31:0]. AES Key (AES_SELKEY = 0)	32'b0
CRYPTO AES TEXTIN register 0 (AES_TEXTIN0)				

Bit	R/W	Symbol	Description	Reset Value
Address :C001 5050h				
[31:0]	R/W	CPU_AES_TEXTIN	CPU_AES_TESTIN[127:96]. AES Input Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO AES TEXTIN register 1 (AES_TEXTIN1)				
Address :C001 5054h				
[31:0]	R/W	CPU_AES_TEXTIN	CPU_AES_TESTIN[95:64]. AES Input Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO AES TEXTIN register 2 (AES_TEXTIN2)				
Address :C001 5058h				
[31:0]	R/W	CPU_AES_TEXTIN	CPU_AES_TESTIN[63:32]. AES Input Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO AES TEXTIN register 3 (AES_TEXTIN3)				
Address :C001 505Ch				
[31:0]	R/W	CPU_AES_TEXTIN	CPU_AES_TESTIN[31:0]. AES Input Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO AES TEXTOUT register 0 (AES_TEXTOUT0)				
Address :C001 5060h				
[31:0]	R/W	CPU_AES_TEXTOUT	CPU_AES_TESTOUT[127:96]. AES Result Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO AES TEXTOUT register 1 (AES_TEXTOUT1)				
Address :C001 5064h				
[31:0]	R/W	CPU_AES_TEXTOUT	CPU_AES_TESTOUT[95:64]. AES Result Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO AES TEXTOUT register 2 (AES_TEXTOUT2)				
Address :C001 5068h				
[31:0]	R/W	CPU_AES_TEXTOUT	CPU_AES_TESTOUT[63:32]. AES Result Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO AES TEXTOUT register 3 (AES_TEXTOUT3)				
Address :C001 506Ch				
[31:0]	R/W	CPU_AES_TEXTOUT	CPU_AES_TESTOUT[31:0]. AES Result Vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO DES Control register (DES_CTRL0)				
Address :C001 5070h				
[31:11]	R	RESERVED	Reserved	21'b0
[10:8]	R/W	DES_TMODE	3DES Mode Setting [8] : 1st stage Mode, 0 : Decoding 1 : Encoding [9] : 2nd stage Mode, 0 : Decoding 1 : Encoding [10] : 3rd stage Mode, 0 : Decoding 1 : Encoding	3'b0
[7]	R	RESERVED	Reserved	1'b0
[6]	R/W	DES_SWAPOUT	Enable the DES Output Swap 0 : Not Swap 1 : Masked	1'b0
[5]	R/W	DES_SWAPIN	Enable the DES Input Swap 0 : Not Swap 1 : Masked	1'b0
[4]	R/W	DES_BLKMODE	Specifies the DES Block Mode 0 : ECB 1 : CBC	1'b0
[3]	R/W	DES_DMAMODE	Enable the DES DMA Interface 0 : Disable 1 : Enable	1'b0
[2]	R/W	DES_MODE	Specifies the DES Mode 0 : DES 1 : 3DES	1'b0
[1]	R/W	DES_ENC	Specifies the DES Encoding Mode (Encryption/Decryption)	1'b0

Bit	R/W	Symbol	Description	Reset Value
			0 : Decryption 1 : Modulation	
[0]	R/W	DES_ENB	Enable the DES Mode 0 : Disable 1 : Enable	1'b0
CRYPTO DES INIT vector register 0 (DES_iv0)				
Address :C001 5074h				
[31:0]	R/W	O_CPU_DES_IV	O_CPU_DES_IV[63:32]. DES Initial vector	32'b0
CRYPTO DES INIT vector register 1 (DES_iv1)				
Address :C001 5078h				
[31:0]	R/W	O_CPU_DES_IV	O_CPU_DES_IV[31:0]. DES Initial vector	32'b0
CRYPTO DES KEY register 0_0 (DES_KEY0_0)				
Address :C001 507Ch				
[31:0]	R/W	O_CPU_DES_KEY0	O_CPU_DES_KEY0[63:32]. DES Key(DES and 1st stage of 3DES)	32'b0
CRYPTO DES KEY register 0_1 (DES_KEY0_1)				
Address :C001 5080h				
[31:0]	R/W	O_CPU_DES_KEY0	O_CPU_DES_KEY0[31:0]. DES Key(DES and 1st stage of 3DES)	32'b0
CRYPTO DES KEY register 1_0 (DES_KEY1_0)				
Address :C001 5084h				
[31:0]	R/W	O_CPU_DES_KEY1	O_CPU_DES_KEY1[63:32]. DES Key (2nd stage of 3DES)	32'b0
CRYPTO DES KEY register 1_1 (DES_KEY1_1)				
Address :C001 5088h				
[31:0]	R/W	O_CPU_DES_KEY1	O_CPU_DES_KEY1[31:0]. DES Key (2nd stage of 3DES)	32'b0
CRYPTO DES KEY register 2_0 (DES_KEY2_0)				
Address :C001 508Ch				
[31:0]	R/W	O_CPU_DES_KEY2	O_CPU_DES_KEY2[63:32]. DES Key (3th stage of 3DES)	32'b0
CRYPTO DES KEY register 2_1 (DES_KEY2_1)				
Address :C001 5090h				
[31:0]	R/W	O_CPU_DES_KEY2	O_CPU_DES_KEY2[31:0]. DES Key (3th stage of 3DES)	32'b0
CRYPTO DES TEXTIN register 0 (DES_TEXTIN0)				
Address :C001 5094h				
[31:0]	R/W	CPU_DES_TESTIN	CPU_DES_TESTIN[63:32]. DES Input vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO DES TEXTIN register 1 (DES_TEXTIN1)				
Address :C001 5098h				
[31:0]	R/W	CPU_DES_TESTIN	CPU_DES_TESTIN[31:0]. DES Input vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO DES TEXTOUT register 0 (DES_TEXTOUT0)				
Address :C001 509Ch				
[31:0]	R/W	CPU_DES_TESTOUT	CPU_DES_TESTOUT[63:32]. DES Input vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO DES TEXTOUT register 1 (DES_TEXTOUT1)				
Address :C001 50A0h				
[31:0]	R/W	CPU_DES_TESTOUT	CPU_DES_TESTOUT [31:0]. DES Input vector in PIO mode (Not DMA Mode)	32'b0
CRYPTO DMA BDMAR register (BDMAR)				
Address :C001 50A4h				

Bit	R/W	Symbol	Description	Reset Value
[31:0]	W	REG_CRT_BDMAR	DMA Access register for AES, DES Input Vectors.	
CRYPTO DMA BDMAW register (BDMAW)				
Address :C001 50A8h				
[31:0]	R	REG_CRT_BDMAW	DMA Access register for AES, DES Output Vectors.(Result Vector)	
CRYPTO DMA HDMAR register (HDMAR)				
Address :C001 50ACh				
[31:0]	W	REG_CRT_BDMAR	DMA Access register for HASH Input Vectors	
CRYPTO HASH Control register 0 (HASH_CTRL0)				
Address :C001 50B0h				
[31:7]	R	RESERVED	Reserved	25'b0
[6:5]	R/W	HASH_INSRC	Specifies the Input of HASH 0: AES input share 1: DES input share 2: HRDMA 3: BWDMA	3'b0
[4]	R/W	HASH_SWAPIN	Enable the HASH Input Swap 0: Not Swap 1: Masked	1'b0
[3]	R/W	HASH_MODE	Specifies the HASH Mode 0: SHA1 1. MD5	1'b0
[2]	R	RESERVED	Reserved	1'b0
[1]	R/W	HASH_DMAMODE	Enable the HASH DMA Interface 0: Disable 1: Enable	1'b0
[0]	R/W	HASH_ENB	Enable the HASH Mode 0: Disable 1: Enable	1'b0
CRYPTO HASH INIT TABLE register 0 (HASH_iv0)				
Address :C001 50B4h				
[31:0]	R/W	O_CPU_HASH_IV	O_CPU_HASH_IV[159:128]. HASH Initial table	32'b0
CRYPTO HASH INIT table register 1 (HASH_iv1)				
Address :C001 50B8h				
[31:0]	R/W	O_CPU_HASH_IV	O_CPU_HASH_IV[127:96]. HASH Initial table	32'b0
CRYPTO HASH INIT table register 2 (HASH_iv2)				
Address :C001 50BCh				
[31:0]	R/W	O_CPU_HASH_IV	O_CPU_HASH_IV[95:64]. HASH Initial table	32'b0
CRYPTO HASH INIT table register 3 (HASH_iv3)				
Address :C001 50C0h				
[31:0]	R/W	O_CPU_HASH_IV	O_CPU_HASH_IV[63:32]. HASH Initial table	32'b0
CRYPTO HASH INIT table register 4 (HASH_iv4)				
Address :C001 50C4h				
[31:0]	R/W	O_CPU_HASH_IV	O_CPU_HASH_IV[31:0]. HASH Initial table	32'b0
CRYPTO HASH TEXTOUT register 0 (HASH_TEXTOUT0)				
Address :C001 50C8h				
[31:0]	R/W	CPU_HASH_TEXTOUT	CPU_HASH_TESTOUT[159:128]. HASH result output	32'b0
CRYPTO HASH TEXTOUT register 1 (HASH_TEXTOUT1)				
Address :C001 50CCh				

Bit	R/W	Symbol	Description	Reset Value
[31:0]	R/W	CPU_HASH_TEXTOUT	CPU_HASH_TESTOUT[127:96]. HASH result output	32'b0
CRYPTO HASH TEXTOUT register 2 (HASH_TEXTOUT2)				
Address :C001 50D0h				
[31:0]	R/W	CPU_HASH_TEXTOUT	CPU_HASH_TESTOUT[96:64]. HASH result output	32'b0
CRYPTO HASH TEXTOUT register 3 (HASH_TEXTOUT3)				
Address :C001 50D4h				
[31:0]	R/W	CPU_HASH_TEXTOUT	CPU_HASH_TESTOUT[63:32]. HASH result output	32'b0
CRYPTO HASH TEXTOUT register 4 (HASH_TEXTOUT4)				
Address :C001 50D8h				
[31:0]	R/W	CPU_HASH_TEXTOUT	CPU_HASH_TESTOUT[31:0]. HASH result output	32'b0
CRYPTO HASH TEXTIN register 0(HASH_TEXTIN)				
Address :C001 50DCh				
[31:0]	R/W	CPU_HASH_TEXTIN	CPU_HASH_TESTIN[127:0]. For DMA and PIO mode	32'b0
CRYPTO HASH TMSG SIZE register 0 (HASH_MSG_SIZE)				
Address :C001 50E0h				
[31:0]	R/W	CPU_HASH_MSGSIZE	CPU_HASH_MSGSIZE[63:32]. HASH result output	32'b0
CRYPTO HASH TMSG SIZE register 1 (HASH_MSG_SIZE)				
Address :C001 50E4h				
[31:0]	R/W	CPU_HASH_MSGSIZE	CPU_HASH_MSGSIZE[31:0]. HASH result output	32'b0

Section 43. Electrical Characteristics

43.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Min	Max	Unit
DC Supply Voltage	VDD	1.0V VDD	-0.5	1.5	V
		1.8V VDD	-0.5	2.5	V
		2.5V VDD	-0.5	3.6	V
		3.3V VDD	-0.5	3.8	V
DC Input Voltage	VIN	1.8V Input Buffer	-0.5	2.5	V
		2.5V Input Buffer	-0.5	3.6	V
		3.3V Input Buffer	-0.5	3.8	V
DC Output Voltage	VOUT	1.8V Output Buffer	-0.5	2.5	V
		2.5V Output Buffer	-0.5	3.6	V
		3.3V Output Buffer	-0.5	3.8	V
DC In/Out Current	linout	-	-20	20	mA
Storage Temperature	Tsa	-	-50 to 150		C

43.2 Recommended Operating Conditions

Pin Name / Symbol	Description	Min	Typical	Max	Unit	Note
VDDI_ARM	DC supply voltage for Cortex-A9 CPU	0.95	1.0	1.05	V	ARM Speed : TBD
		1.045	1.1	1.155	V	ARM Speed : TBD
		1.14	1.2	1.26	V	ARM Speed : TBD
		1.235	1.3	1.365	V	ARM Speed : TBD
VDDI	DC supply voltage for CORE	0.95	1.0	1.05	V	-
VDDP18	DC supply voltage for 1.8V Internal IO	1.7	1.8	1.9	V	-
DVDD33_IO	DC supply voltage for 3.3V IO	3.0	3.3	3.6	V	-
VDDQ	DC supply voltage for DRAM IP (LPDDR2)	1.14	1.2	1.26	V	-
	DC supply voltage for DRAM IP (LPDDR3)	1.14	1.2	1.26	V	-
	DC supply voltage for DRAM IP (1.35V DDR3)	1.283	1.35	1.417	V	-
	DC supply voltage for DRAM IP (1.5V DDR3)	1.425	1.5	1.575	V	-
VDDI10_ALIVE	DC supply voltage for CORE ALIVE	0.95	1.0	1.05	V	-
VDDP18_ALIVE	DC supply voltage for 1.8V Internal IO ALIVE	1.7	1.8	1.9	V	-
VDD33_ALIVE	DC supply voltage for 3.3V ALIVE	3.0	3.3	3.6	V	-
DVDD10_USB0	DC supply voltage for 1.0V USB OTG	0.95	1.0	1.05	V	-
VDD18_USB0	DC supply voltage for 1.8V USB OTG	1.7	1.8	1.9	V	-
VDD33_USB0	DC supply voltage for 3.3V USB OTG	3.0	3.3	3.6	V	-
DVDD10_USBHOST0	DC supply voltage for 1.0V USB HOST	0.95	1.0	1.05	V	-
VDD18_USBHOST	DC supply voltage for 1.8V USB HOST	1.7	1.8	1.9	V	-
VDD33_USBHOST	DC supply voltage for 3.3V USB HOST	3.0	3.3	3.6	V	-
DVDD12_HSIC	DC supply voltage for 1.2V USB HSIC HOST	1.15	1.2	1.25	V	1)*
VDD18_RTC	DC supply voltage for 1.8V RTC Crystal	1.7	1.8	1.9	V	-
VDD18_OSC	DC supply voltage for 1.8V PLL Crystal	1.7	1.8	1.9	V	-
AVDD10_LV	DC supply voltage for 1.0V LVDS	0.95	1.0	1.05	V	1)*
AVDD18_LV	DC supply voltage for 1.8V LVDS	1.71	1.8	1.89	V	1)*
AVDD10_HM	DC supply voltage for 1.0V HDMI	0.95	1.0	1.05	V	1)*
VDD10_HM_PLL	DC supply voltage for 1.0V HDMI PLL	0.95	1.0	1.05	V	1)*
VDD18_HM	DC supply voltage for 1.8V HDMI	1.71	1.8	1.89	V	1)*
M_VDD10	DC supply voltage for 1.0V MIPI	0.95	1.0	1.05	V	1)*
M_VDD10_PLL	DC supply voltage for 1.0V MIPI PLL	0.95	1.0	1.05	V	1)*
M_VDD18	DC supply voltage for 1.8V MIPI	1.7	1.8	1.9	V	1)*
AVDD18_ADC	DC supply voltage for 1.8V ADC	1.7	1.8	1.9	V	-
ADCREF	Reference 1.8V for ADC	1.7	1.8	1.9	V	-
AVDD18_PLL	DC supply voltage for 1.8V PLL	1.7	1.8	1.9	V	-
DVDD_VID0	DC supply voltage for 2.8V VID0	1.7	2.8	3.6	V	-
DVDD_VID2_SD2	DC supply voltage for 2.8V VID2/SD2	1.7	2.8	3.6	V	-

DVDD_GMAC	DC supply voltage for 2.8V Ethernet MAC	1.7	2.8	3.6	V	1)*
T _a	Operating Ambient Temperature	TBD		C		

1)* : This power pin can be tied to GND when this function is not used

43.3 D.C. Electrical Characteristics

DC electrical specification for 3.3V tol

VDD=1.65V~3.60V, Vext=3.0~3.6V, Tj=-40 to 125C

Parameter		Condition		Min	Typ	Max	Unit
Vtol	Tolerant external voltage**	VDD Power Off & On				3.6	V
Vih	High Level Input Voltage						
	CMOS Interface			0.7VDD		VDD+0.3	V
Vil	Low Level Input Voltage						
	CMOS Interface	VDD=2.5V±10%, 3.3V±10%	-0.3		0.7	0.3VDD	V
ΔV		VDD=1.8V±10%	-0.3				
Hysteresis Voltage		0.15				V	
Iih	High Level Input Current						
	Input Buffer	Vin=VDD	VDD Power ON	-3		3	uA
			VDD Power Off & SNS=0	-5		5	uA
	Input Buffer with pull-down	Vin=VDD	VDD=3.3V±10%	15	40	80	uA
			VDD=2.5V±10%	15	40	80	uA
			VDD=1.8V±10%	15	40	80	uA
Iil	Low Level Input Current						
	Input Buffer	Vin=VSS	VDD Power ON & Off	-3		3	uA
	Input Buffer with pull-down	Vin=VSS	VDD=3.3V±10%	-15	-40	-110	uA
			VDD=2.5V±10%	-15	-40	-110	uA
			VDD=1.8V±10%	-15	-40	-110	uA
Voh	Output High Voltage	Ioh = -1.8mA, -3.6mA, -7.2mA, -10.8mA		0.8VDD		VDD	V
Vol	Output Low Voltage	Ioh = -1.8mA, -3.6mA, -7.2mA, -10.8mA		0		0.2VDD	V
Ioz	Output Hi-Z Current			-5		5	uA
CIN	Input Capacitance	Any input and Bidirectional buffers				5	pF
COUT	Output capacitance	Any output buffer				5	pF

** specification is only available on tolerant cells