GM8125/GM8126/GM8128

# H.264 IP CAM SOC

Data Sheet
Rev.: 0.4
Issue Date: June 2011

# REVISION HISTORY

## GM8125/GM8126/GM8128 Data Sheet

| Date | Rev. | From | To |
|---|---|---|---|
| Sept. 2010 | 0.1 | - | Original |
| Feb. 2011 | 0.2 | - | Added the MP Chip specification |
| Mar. 2011 | 0.3 | - | Updated the specification |
| Jun. 2011 | 0.4 | - | • Updated the description of Operating Voltage in Section 1.2<br>• Updated Table 2-1, Table 2-2, Table 4-21, Table 32-9, and Table 33-2 |

Copyright © 2011 Grain Media, Inc.

All Rights Reserved.

Printed in Taiwan 2011

Grain Media and the Grain Media Logo are trademarks of Grain Media, Inc. in Taiwan and/or other countries.
Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support application where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Grain Media's product specification or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Grain Media or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will Grain Media be liable for damages arising directly or indirectly from any use of the information contained in this document.

Grain Media, Inc.
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:
http://www.grain-media.com

# TABLE OF CONTENTS


Chapter 1 Introduction ..... 1

1.1 Version of the Chip ..... 2

1.2 Features ..... 2

1.3 Block Diagram ..... 8

1.3.1 FA626 ..... 9

1.3.2 AHB Controller (AHBC) ..... 9

1.3.3 DDR2 SDRAM Controller (DDRC) ..... 9

1.3.4 Static Memory Controller (SMC) ..... 10

1.3.5 NAND Flash Controller (NANDC) ..... 10

1.3.6 Direct Memory Access Controller (DMAC) ..... 10

1.3.7 USB 2.0 OTG Controller ..... 11

1.3.8 H.264 Encoder ..... 12

1.3.9 MPEG4/JPEG Engine ..... 12

1.3.10 3D De-interlace De-noise Filter ..... 13

1.3.11 Video Capture ..... 14

1.3.12 Image Sensor Processor ..... 15

1.3.13 LCD Controller (LCDC) ..... 16

1.3.14 10/100 Mbps Ethernet Controller (MAC) ..... 16

1.3.15 AES/DES/TDES Cipher Controller ..... 17

1.3.16 AHB-to-APB Bridge ..... 18

1.3.17 Timer ..... 18

1.3.18 Pulse Width Modulation (PWM) ..... 18

1.3.19 Watchdog Timer (WDT) ..... 19

1.3.20 Interrupt Controller (INTC) ..... 19

1.3.21 GPIO ..... 20

1.3.22 I²C Bus ..... 20

1.3.23 Power Management Unit (PMU) ..... 21

1.3.24 $I^2S$ Controller ..... 21

1.3.25 UART Controller ..... 22
1.3.26 Secure Digital Card (SDC) Controller ..... 22
1.3.27 Comparsion Table of GM8125/GM8126/GM8128 ..... 23
1.3.28 Comparsion Table of E-PAD TQFP-176 and TFBGA-256 ..... 23
Chapter 2 Signal Description ..... 25
2.1 Signal Description ..... 26
2.2 TFBGA-256 Pin Assignments ..... 36
Chapter 3 Memory Map ..... 37
3.1 Memory Map ..... 38
Chapter 4 Power Management Unit ..... 39
4.1 General Description ..... 40
4.2 Features ..... 40
4.3 Clock Manager ..... 41
4.3.1 32.768-kHz Oscillator ..... 42
4.3.2 Core PLL (PLL1) ..... 42
4.3.3 Peripheral PLL (PLL2) ..... 43
4.3.4 Clock Gating ..... 43
4.3.5 Hardware Reset ..... 44
4.3.6 Watchdog Reset ..... 44
4.3.7 Normal Mode ..... 44
4.3.8 IDLE Mode ..... 45
4.3.9 Standby Mode ..... 45
4.3.10 Sleep Mode ..... 46
4.3.11 Frequency Change Sequence (FCS) ..... 46
4.4 Programming Model ..... 48
4.4.1 Summary of Clock and Power Manager Registers ..... 48
4.4.2 Register Descriptions ..... 49
Chapter 5 FA626TE ..... 107
5.1 General Description ..... 108
5.2 Block Diagram ..... 109
5.2.1 CPU Core ..... 109
5.2.2 Branch Prediction Unit (BPU) ..... 110
5.2.3 Instruction Cache (ICache) ..... 110
5.2.4 Data Cache (DCache) ..... 110

5.2.5 Memory Management Unit (MMU) ..... 111
5.2.6 IScratchpad and DScratchpad ..... 111
5.2.7 Bus Interface Unit (BIU) ..... 111
5.2.8 Write Buffer (WB) ..... 112
5.2.9 ICE ..... 112
5.2.10 Power-saving Control Unit ..... 112
5.2.11 Coprocessor Transfer Instruction ..... 112
Chapter 6 AHB Controller ..... 113
6.1 General Description ..... 114
6.2 Features ..... 114
6.3 Block Diagram ..... 115
6.4 Memory Map/Register Definition ..... 115
6.4.1 Summary of Control Registers ..... 115
6.4.2 AHB Slave *n* Base/Size Register ..... 116
6.4.3 Priority Control Register ..... 117
6.4.4 Idle Count Register ..... 117
6.4.5 Control Register ..... 117
6.4.6 AHB Bus Request Enable Register ..... 119
6.5 Functional Description ..... 119
6.5.1 Arbiter ..... 119
6.5.2 Decoder ..... 119
6.6 Multiplexer ..... 120
6.7 Register Slave ..... 120
Chapter 7 APB Bridge ..... 121
7.1 General Description ..... 122
7.2 Features ..... 122
7.3 Block Diagram ..... 123
7.4 DMA Routing Table ..... 123
7.5 Memory Map/Register Definition ..... 124
7.5.1 Base/Size Register of APB Slave *n* ..... 125
7.5.2 Source Addresses of DMA Channels A/B/C/D (Optional for DMA) ..... 126
7.5.3 Destination Addresses of DMA Channels A/B/C/D (Optional for DMA) ..... 126
7.5.4 Cycle of DMA Channels A/B/C/D (Optional for DMA) ..... 126
7.5.5 Command of DMA Channels A/B/C/D (Optional for DMA) ..... 127

7.5.6 APB Control Register ..... 129
7.5.7 Revision Register ..... 130
7.6 Functional Description ..... 130
7.6.1 Control Register ..... 130
7.6.2 AHB Slave for Accessing APB Devices ..... 130
7.6.3 AHB Slave for Accessing APB Bridge Registers ..... 131
7.6.4 AHB Master ..... 131
7.6.5 APB Master ..... 131
7.6.6 DMA Engine ..... 131
7.6.7 DMA Hardware Handshake Mode ..... 132
7.6.8 Arbiter ..... 132
Chapter 8 Static Memory Controller ..... 133
8.1 Features ..... 134
8.2 Block Diagram ..... 134
8.2.1 AHB Slave ..... 135
8.2.2 Control Engine ..... 135
8.3 Connection of Different Device Types ..... 135
8.3.1 Asynchronous Devices ..... 136
8.4 Programming Model ..... 136
8.4.1 Configuration Register of Memory Bank *n* (Offset = n * 8) ..... 137
8.4.2 Timing Parameter Register of Memory Bank (Offset = (*n* * 8) + 4) ..... 139
8.4.3 Asynchronous Devices Timing Waveform ..... 140
8.4.4 Timing Waveform of Variable Latency I/O (VLIO) Devices ..... 143
8.4.5 Shadow Status Register (Offset = 0x40) ..... 144
Chapter 9 DDR2 Memory Controller ..... 147
9.1 General Description ..... 148
9.2 Features ..... 148
9.3 Block Diagram ..... 149
9.4 Function Description ..... 150
9.4.1 Modes of Operation ..... 150
9.4.2 Refresh Function ..... 150
9.4.3 Channel Arbitration ..... 150
9.5 Programming Model ..... 155
9.5.1 Register Descriptions ..... 156

9.5.2 Memory Controller Configuration Register (Offset = 0x00) ..... 157
9.5.3 Memory Controller State Control Register (Offset = 0x04) ..... 159
9.5.4 Mode Register Set Value Register of MR and EMR (Offset = 0x08) ..... 161
9.5.5 Mode Register Set Value Register of EMR2 and EMR3 (Offset = 0x0C) . 161
9.5.6 External Rank 0/1 Register (Offset = 0x10) ..... 162
9.5.7 Timing Parameter 0 Register (Offset = 0x14) ..... 164
9.5.8 Timing Parameter 1 Register (Offset = 0x18) ..... 165
9.5.9 Timing Parameter 2 Register (Offset = 0x1C) ..... 167
9.5.10 DDR2 PHY Command and Data Block Control Register (Offset = 0x20) . 169
9.5.11 DDR PHY Read Path DLL Delay Tuning Register (Offset = 0x24) ..... 172
9.5.12 COMPBLK Control Register (Offset = 0x28) ..... 173
9.5.13 Automatic Power-down Control Register (Offset = 0x2C) ..... 173
9.5.14 Channel Arbitration Setup Register (Offset = 0x30) ..... 174
9.5.15 Channel Arbiter Grant Count Register – A (Offset = 0x34) ..... 174
9.5.16 Channel Arbiter Grant Count Register – B (Offset = 0x38) ..... 176
9.5.17 Command Flush Control Register (Offset = 0x40) ..... 177
9.5.18 Command Flush Status Register (Offset = 0x44) ..... 178
9.5.19 AHB SPLIT Control Register (Offset = 0x48) ..... 178
9.5.20 AHB INCR Read Pre-fetch Length 1 (Offset = 0xA0) ..... 179
9.5.21 AHB INCR Read Pre-fetch Length 2 (Offset = 0xA4) ..... 181
9.5.22 Initialization of Waiting Cycle Count 1 (Offset = 0xA8) ..... 182
9.6 Memory Address Table (MA Table) ..... 182
Chapter 10 DMA Controller ..... 197
10.1 General Description ..... 198
10.2 Features ..... 198
10.3 Block Diagram ..... 199
10.3.1 AHB Master Interface ..... 199
10.3.2 AHB Slave Interface ..... 200
10.3.3 FIFO Buffer ..... 200
10.3.4 DMA Core ..... 200
10.4 Programming Model ..... 200
10.4.1 Prioritizing Arbiter ..... 200
10.4.2 Chain Transfer ..... 201
10.4.3 DMA Hardware Handshake Mode ..... 205

10.4.4 DMA Normal Mode ..... 209
10.4.5 Summary of DMA Controller Registers ..... 209
10.5 Programming Sequence ..... 235
10.5.1 Channel Initialization (Normal Mode, No Chain Transfer) ..... 235
10.5.2 Channel Initialization (Normal Mode, Chain Transfer) ..... 236
10.5.3 Channel Initialization (Hardware Handshake Mode, No Chain Transfer) ..... 237
10.5.4 Channel Initialization (Hardware Handshake Mode, Chain Transfer) ..... 238
10.5.5 Channel Abort (During DMA Data Transfer) ..... 238
Chapter 11 AES-DES Cipher Coprocessor ..... 239
11.1 General Description ..... 240
11.1.1 Terminology ..... 240
11.2 Features ..... 241
11.3 Block Diagram ..... 242
11.4 Function Description ..... 242
11.4.1 AHB Slave and Control Register ..... 242
11.4.2 AHB Master ..... 242
11.4.3 DMA Engine ..... 243
11.4.4 Data FIFO ..... 243
11.4.5 Security Engine ..... 243
11.5 Register Descriptions ..... 244
11.5.1 Summary of Control Registers ..... 244
11.5.2 Register Descriptions ..... 245
11.6 Programming Sequence ..... 252
Chapter 12 USB 2.0 OTG Controller ..... 257
12.1 General Description ..... 258
12.2 Features ..... 258
12.3 Block Diagram ..... 259
12.4 Register Definition ..... 260
12.4.1 Register Summary ..... 260
Chapter 13 H.264 Encoder ..... 305
13.1 General Description ..... 306
13.2 Features ..... 306
13.3 Block Diagram ..... 307
13.3.1 Video Input Format ..... 307

13.3.2 AHB Host Interface ..... 307
13.3.3 DMA ..... 307
13.3.4 Motion Estimation ..... 308
13.3.5 Intra Prediction ..... 308
13.3.6 Entropy Coding ..... 308
13.3.7 TQ and IQIT ..... 308
13.3.8 Deblocking Filter ..... 308
13.3.9 System Controller ..... 309
Chapter 14 MPEG/JPEG Engine ..... 311
14.1 General Description ..... 312
14.2 Features ..... 312
14.3 Block Diagram ..... 314
14.3.1 AHB Interface ..... 315
14.3.2 DMA ..... 315
14.3.3 Motion Estimation ..... 315
14.3.4 DCT/IDCT ..... 315
14.3.5 Quantization/Inverse Quantization ..... 315
14.3.6 AC/DC Prediction ..... 316
14.3.7 Zigzag Scan ..... 316
14.3.8 Variable Length Coding/Decoding ..... 316
14.3.9 Motion Compensation ..... 316
14.3.10 Local Memory Controller ..... 317
Chapter 15 LCD Controller ..... 319
15.1 General Description ..... 320
15.2 Features ..... 320
15.3 Block Diagram ..... 322
15.4 Function Description ..... 323
15.4.1 FIFO Controllers and FIFO ..... 323
15.4.2 Pixel Data Unpack ..... 323
15.4.3 Data Mode ..... 327
15.4.4 On-screen Display (Simple OSD) ..... 330
15.4.5 Scaler ..... 332
15.4.6 Interrupt Controller ..... 335
15.5 Programming Model ..... 335

15.5.1 Summary of Control Registers ..... 335
15.5.2 LCD Global Parameters ..... 337
15.5.3 LCD Timing and Polarity Parameters ..... 345
15.5.4 LCD Output Format Parameters ..... 348
15.5.5 Scaler Control Registers ..... 353
15.5.6 Virtual Screen ..... 357
15.5.7 OSD Control Registers for Simple OSD ..... 358
15.5.8 OSD Font Database Write Accessing Port (Offset 0x8000 ~ 0xBFFC) ..... 361
15.5.9 OSD Window Attribute Write Accessing Port (Offset 0xC000 ~ 0xC7FC) 361
Chapter 16 10/100 Ethernet Controller ..... 363
16.1 General Description ..... 364
16.2 Features ..... 364
16.3 Interface Clocking ..... 365
16.4 Register Definition ..... 365
16.4.1 Interrupt Status Register, ISR, 32'h0 (Offset: 0x00 ~ 0x03) ..... 365
16.4.2 Interrupt Enable Register, IME, 32'h0 (Offset: 0x04 ~ 0x07) ..... 366
16.4.3 MAC Most Significant Address Register, MAC_MADR, 32'h0 (Offset: 0x08 ~ 0x0B) ..... 366
16.4.4 MAC Least Significant Address Register, MAC_LADR, 32'h0 (Offset: 0x0C ~ 0x0F) ..... 366
16.4.5 Multicast Address Hash Table 0 Register, MAHT0, 32'h0 (Offset: 0x10 ~ 0x13) ..... 367
16.5 Multicast Address Hash Table 1 Register, MAHT1, 32'h0 (Offset: 0x14 ~ 0x17) .. 367
16.5.1 Transmit Poll Demand Register, TXPD, 32'h0 (Offset: 0x18 ~ 0x1B) ..... 367
16.5.2 Receive Poll Demand Register, RXPD, 32'h0 (Offset: 0x1C ~ 0x1F) ..... 367
16.5.3 Transmit Ring Base Address Register, TXR_BADR, 32'h0 (Offset: 0x20 ~ 0x23) ..... 368
16.5.4 Receive Ring Base Address Register, RXR_BADR, 32'h0 (Offset: 0x24 ~ 0x27) ..... 368
16.5.5 Interrupt Timer Control Register, ITC, 32'h0 (Offset: 0x28 ~ 0x2B) ..... 368
16.5.6 Automatic Polling Timer Control Register, APTC, 32'h0, (Offset: 0x2C ~ 0x2F) ..... 371
16.5.7 DMA Burst Length and Arbitration Control Register, DBLAC, 32'h0, (Offset: 0x30 ~ 0x33) ..... 373
16.5.8 Revision Register, REVR, 32'h0, (Offset: 0x34 ~ 0x37) ..... 375
16.5.9 Feature Register, FEAR, 32'h0, (Offset: 0x38 ~ 0x3B) ..... 375

16.5.10 Reserved Register, (Offset: 0x3C ~ 0x87) ..... 376
16.5.11 MAC Control Register, MACCR, 32'h0, (Offset: 0x88 ~ 0x8B) ..... 376
16.5.12 MAC Status Register, MACSR, 32'h0, (Offset: 0x8C ~ 0x8F) ..... 377
16.5.13 PHY Control Register, PHYCR, 32'h0, (Offset: 0x90 ~ 0x93) ..... 377
16.5.14 PHY Write Data Register, PHYWDATA, 32'h0, (Offset: 0x94 ~ 0x97) ..... 378
16.5.15 Flow Control Register, FCR, 32'h0000_A400, (Offset: 0x98 ~ 0x9B) ..... 378
16.5.16 Back Pressure Register, BPR, 32'h0000_0400, (Offset: 0x9C ~ 0x9F) .... 379
16.5.17 Wake-On-LAN Control Register, WOLCR, 32'h0, (Offset: 0xA0 ~ 0xA3) . 380
16.5.18 Wake-On-LAN Status Register, WOLSR, 32'h0, (Offset: 0xA4 ~ 0xA7) ... 381
16.5.19 Wake-up Frame CRC Register, WFCRC, 32'h0, (Offset: 0xA8 ~ 0xAB) .. 381
16.5.20 Wake-up Frame Byte Mask 1st Double Word Register, WFBM1, 32'h0, (Offset: 0xB0 ~ 0xB3) ..... 382
16.5.21 Wake-up Frame Byte Mask 2nd Double Word Register, WFBM2, 32'h0, (Offset: 0xB4 ~ 0xB7) ..... 383
16.5.22 Wake-up Frame Byte Mask 3rd Double Word Register, WFBM3, 32'h0, (Offset: 0xB8 ~ 0xBB) ..... 384
16.5.23 Wake-up Frame Byte Mask 4th Double Word Register, WFBM4, 32'h0, (Offset: 0xBC ~ 0xBF) ..... 385
16.5.24 Test Seed Register, TS, 32'h0, (Offset: 0xC4 ~ 0xC7) ..... 385
16.5.25 DMA/FIFO State Register, DMAFIFOS, 32'h0, (Offset: 0xC8 ~ 0xCB) ..... 385
16.5.26 Test Mode Register, TM, 32'h0, (Offset: 0xCC ~ 0xCF) ..... 386
16.5.27 TX_MCOL and TX_SCOL Counter Register, 32'h0, (Offset: 0xD4 ~ 0xD7) 387
16.5.28 RPF and AEP Counter Register, 32'h0, (Offset: 0xD8 ~ 0xDB) ..... 387
16.5.29 XM and PG Counter Register, 32'h0, (Offset: 0xDC ~ 0xDF) ..... 387
16.5.30 RUNT_CNT and TLCC Counter Register, 32'h0, (Offset: 0xE0 ~ 0xE3) .. 388
16.5.31 CRCER_CNT and FTL_CNT Counter Register, 32'h0, (Offset: 0xE4 ~ 0xE7) ..... 388
16.5.32 RLC and RCC Counter Register, 32'h0, (Offset: 0xE8 ~ 0xEB) ..... 388
16.5.33 BROC Counter Register, 32'h0, (Offset: 0xEC ~ 0xEF) ..... 388
16.5.34 MULCA Counter Register, 32'h0, (Offset: 0xF0 ~ 0xF3) ..... 389
16.5.35 RP Counter Register, 32'h0, (Offset: 0xF4 ~ 0xF7) ..... 389
16.5.36 XP Counter Register, 32'h0, (Offset: 0xF8 ~ 0xFB) ..... 389
16.6 Function Description ..... 389
16.6.1 Half-duplex (CSMA/CD Access Protocol) ..... 389
16.6.2 Full-duplex Ethernet ..... 390
16.6.3 Loop Back ..... 391

16.6.4 Transmit Descriptors and Data Buffers ..... 392
16.6.5 Receive Descriptors and Data Buffers ..... 394
16.6.6 Transmitting Packets ..... 397
16.6.7 Receiving Packets ..... 397
16.6.8 Zero-Copy ..... 398
16.6.9 Ethernet Address Filtering ..... 398
16.6.10 DMA Arbitration Scheme ..... 400
16.6.11 Wake-On-LAN ..... 400
16.6.12 Link Status Change ..... 400
16.6.13 Magic Packet ..... 401
16.6.14 Wake-up Frame ..... 401
16.6.15 Power-down Mode ..... 403
16.6.16 Flow Control ..... 403
16.7 Initialization/Application Information ..... 404
16.7.1 Frame Transmitting Procedure ..... 404
16.7.2 Frame Receiving Procedure ..... 406
16.7.3 Procedures of Entering to and Exiting from Power-down Mode ..... 407
16.7.4 Ethernet Frame Formats ..... 408
16.8 MII Management Interface ..... 409
16.9 Multicast Address Hash Table Filtering ..... 410
Chapter 17 TV Controller ..... 413
17.1 General Description ..... 414
17.2 Features ..... 414
17.3 Block Diagram ..... 415
17.4 Programming Model ..... 415
17.4.1 Summary of FTTVE100_S Controller Registers ..... 415
Chapter 18 3D-deinterlace De-noise Filter ..... 421
18.1 General Description ..... 422
18.2 Features ..... 422
18.3 Block Diagram ..... 423
18.4 Programming Model ..... 423
18.4.1 Control Register Summary ..... 423
18.4.2 Register Description ..... 425
Chapter 19 Video Capture ..... 437

19.1 General Description ..... 438
19.2 Features ..... 438
19.3 Block Diagram ..... 439
19.4 Memory Map/Register Definition ..... 439
19.4.1 Mapping of Usual Special-purpose Registers ..... 439
19.4.2 Special-purpose Registers and Functions ..... 444
19.4.3 P1SHARPEN (Offset 0x00D8) ..... 449
19.4.4 P1SIZE0 (Offset 0x0020) ..... 449
19.4.5 P1SIZE1 (Offset 0x0024) ..... 449
19.4.6 P1SIZE2 (Offset 0x0028) ..... 450
19.4.7 DFORMAT Register (Offset 0x0038) ..... 450
19.4.8 GBATRB (Offset 0x003C) ..... 451
19.4.9 FMRATE (Offset 0x0040) ..... 453
19.4.10 SRCIF (Offset 0x0044) ..... 455
19.4.11 SRCSIZE0 (Offset 0x0050) ..... 456
19.4.12 P0SIZE3 (Offset 0x0054) ..... 456
19.4.13 P0SIZE4 (Offset 0x0058) ..... 456
19.4.14 P0SIZE5 (Offset 0x005C) ..... 457
19.4.15 SRCSIZE1 (Offset 0x0060) ..... 457
19.4.16 P1SIZE3 (Offset 0x0064) ..... 458
19.4.17 P1SIZE4 (Offset 0x0068) ..... 458
19.4.18 P1SIZE5 (Offset 0x006C) ..... 459
19.4.19 P1SHARPEN (Offset 0x00D8) ..... 459
19.4.20 P0MSK0COR (Offset 0x0100) ..... 459
19.4.21 P0MSK0SIZE0 (Offset 0x0104) ..... 460
19.4.22 P0MSK0SIZE1 (Offset 0x0108) ..... 460
19.4.23 P0MSK1COR (Offset 0x010C) ..... 462
19.4.24 P0MSK1SIZE0 (Offset 0x0110) ..... 462
19.4.25 P0MSK1SIZE1 (Offset 0x0114) ..... 463
19.4.26 P0MSK2COR (Offset 0x0118) ..... 463
19.4.27 P0MSK2SIZE0 (Offset 0x011C) ..... 464
19.4.28 P0MSK2SIZE1 (Offset 0x0120) ..... 464
19.4.29 P0MSK1COR (Offset 0x0124) ..... 464
19.4.30 P0MSK3SIZE0 (Offset 0x0128) ..... 465

19.4.31 P0MSK3SIZE1 (Offset 0x012C) ..... 465
19.4.32 P0MSK4COR (Offset 0x0130) ..... 466
19.4.33 P0MSK4SIZE0 (Offset 0x0134) ..... 466
19.4.34 P0MSK4SIZE1 (Offset 0x0138) ..... 467
19.4.35 P0MSK5COR (Offset 0x013C) ..... 467
19.4.36 P0MSK5SIZE0 (Offset 0x0140) ..... 468
19.4.37 P0MSK5SIZE1 (Offset 0x0144) ..... 468
19.4.38 P0MSK6COR (Offset 0x0148) ..... 468
19.4.39 P0MSK6SIZE0 (Offset 0x014C) ..... 469
19.4.40 P0MSK6SIZE1 (Offset 0x0150) ..... 469
19.4.41 P0MSK6COR (Offset 0x0154) ..... 470
19.4.42 P0MSK7SIZE0 (Offset 0x0158) ..... 470
19.4.43 P0MSK7SIZE1 (Offset 0x015C) ..... 471
19.4.44 P1MSK0COR (Offset 0x0160) ..... 471
19.4.45 P1MSK0SIZE0 (Offset 0x0164) ..... 472
19.4.46 P1MSK0SIZE1 (Offset 0x0168) ..... 472
19.4.47 P1MSK1COR (Offset 0x016C) ..... 472
19.4.48 P1MSK1SIZE0 (Offset 0x0170) ..... 473
19.4.49 P1MSK1SIZE1 (Offset 0x0174) ..... 473
19.4.50 P1MSK2COR (Offset 0x0178) ..... 474
19.4.51 P1MSK2SIZE0 (Offset 0x017C) ..... 474
19.4.52 P1MSK2SIZE1 (Offset 0x0180) ..... 475
19.4.53 P1MSK1COR (Offset 0x0184) ..... 475
19.4.54 P1MSK3SIZE0 (Offset 0x0188) ..... 476
19.4.55 P1MSK3SIZE1 (Offset 0x018C) ..... 476
19.4.56 P1MSK4COR (Offset 0x0190) ..... 476
19.4.57 P1MSK4SIZE0 (Offset 0x0194) ..... 477
19.4.58 P1MSK4SIZE1 (Offset 0x0198) ..... 477
19.4.59 P1MSK5COR (Offset 0x019C) ..... 478
19.4.60 P1MSK5SIZE0 (Offset 0x01A0) ..... 478
19.4.61 P1MSK5SIZE1 (Offset 0x01A4) ..... 479
19.4.62 P1MSK6COR (Offset 0x01A8) ..... 479
19.4.63 P1MSK6SIZE0 (Offset 0x01AC) ..... 480
19.4.64 P1MSK6SIZE1 (Offset 0x01B0) ..... 480

19.4.65 P1MSK6COR (Offset 0x01B4) ..... 480
19.4.66 P1MSK7SIZE0 (Offset 0x01B8) ..... 481
19.4.67 P1MSK7SIZE1 (Offset 0x01BC) ..... 481
19.4.68 DICTRL0 (Offset 0x01E0) ..... 481
19.4.69 DICTRL1 (Offset 0x001E4) ..... 482
19.4.70 MISC0 (Offset 0x001F0) ..... 482
19.4.71 P0DMA0 (Offset 0x0200) ..... 483
19.4.72 P1DMA0 (Offset 0x0204) ..... 485
19.4.73 P0DMA1 (Offset 0x0218) ..... 486
19.4.74 P1DMA1 (Offset 0x021C) ..... 486
19.4.75 P0MDEST0 (Offset 0x0220) ..... 487
19.4.76 P0MDEST1 (Offset 0x0224) ..... 487
19.4.77 P0MDEST2 (Offset 0x0230) ..... 487
19.4.78 P0MDEST3 (Offset 0x0234) ..... 487
19.4.79 P0MDEST4 (Offset 0x0240) ..... 488
19.4.80 P0MDEST5 (Offset 0x0244) ..... 488
19.4.81 P1MDEST0 (Offset 0x0250) ..... 489
19.4.82 P1MDEST1 (Offset 0x0254) ..... 489
19.4.83 P1MDEST2 (Offset 0x0260) ..... 489
19.4.84 P1MDEST3 (Offset 0x0264) ..... 489
19.4.85 P1MDEST4 (Offset 0x0270) ..... 490
19.4.86 P1MDEST5 (Offset 0x0274) ..... 490
19.4.87 VBICTRL0 (Offset 0x0290) ..... 490
19.4.88 VBICTRL1 (Offset 0x0294) ..... 490
19.4.89 VBICTRL2 (Offset 0x02A0) ..... 491
19.4.90 VBICTRL3 (Offset 0x02B0) ..... 491
19.4.91 P0OSDFONT (Offset 0x0300) ..... 491
19.4.92 P0OSDDISP (Offset 0x0304) ..... 492
19.4.93 P0OSDREAD (Offset 0x0308) ..... 493
19.4.94 P0OSDEN (Offset 0x030C) ..... 493
19.4.95 P0OSDPAT0 (Offset 0x0310) ..... 494
19.4.96 P0OSDPAT1 (Offset 0x0314) ..... 494
19.4.97 P0OSDPALT2 (Offset 0x0318) ..... 494
19.4.98 P0OSDPAT3 (Offset 0x031C) ..... 495

19.4.99 P0OSDPAT4 (Offset 0x0320) ..... 495
19.4.100 P0OSDPAT5 (Offset 0x0324) ..... 495
19.4.101 P0OSDPAT6 (Offset 0x0328) ..... 496
19.4.102 P0OSDCOR0 (Offset 0x0330) ..... 496
19.4.103 P0OSDWSZ0 (Offset 0x0334) ..... 497
19.4.104 P0OSDSSZ0 (Offset 0x0338) ..... 498
19.4.105 P0OSDFSZ0 (Offset 0x033C) ..... 498
19.4.106 P0OSDCOR1 (Offset 0x0340) ..... 499
19.4.107 P0OSDWSZ1 (Offset 0x0344) ..... 500
19.4.108 P0OSDSSZ1 (Offset 0x0348) ..... 501
19.4.109 P0OSDFSZ1 (Offset 0x034C) ..... 501
19.4.110 P0OSDCOR2 (Offset 0x0350) ..... 502
19.4.111 P0OSDWSZ2 (Offset 0x0354) ..... 503
19.4.112 P0OSDSSZ2 (Offset 0x0358) ..... 503
19.4.113 P0OSDFSZ2 (Offset 0x035C) ..... 503
19.4.114 P0OSDCOR3 (Offset 0x0360) ..... 504
19.4.115 P0OSDWSZ3 (Offset 0x0364) ..... 505
19.4.116 P0OSDSSZ3 (Offset 0x0368) ..... 506
19.4.117 P0OSDFSZ3 (Offset 0x036C) ..... 506
19.4.118 P1OSDFONT (Offset 0x0370) ..... 506
19.4.119 P1OSDDISP (Offset 0x0374) ..... 507
19.4.120 P1OSDREAD (Offset 0x0378) ..... 507
19.4.121 P1OSDEN (Offset 0x037C) ..... 507
19.4.122 P1OSDPAT0 (Offset 0x0380) ..... 508
19.4.123 P1OSDPAT1 (Offset 0x0384) ..... 508
19.4.124 P1OSDPALT2 (Offset 0x0388) ..... 509
19.4.125 P1OSDPAT3 (Offset 0x038C) ..... 509
19.4.126 P1OSDPAT4 (Offset 0x0390) ..... 509
19.4.127 P1OSDPAT5 (Offset 0x0394) ..... 510
19.4.128 P1OSDPAT6 (Offset 0x0398) ..... 510
19.4.129 P1OSDCOR0 (Offset 0x03A0) ..... 510
19.4.130 P1OSDWSZ0 (Offset 0x03A4) ..... 511
19.4.131 P1OSDSSZ0 (Offset 0x03A8) ..... 512
19.4.132 P1OSDFSZ0 (Offset 0x03AC) ..... 512

19.4.133 P1OSDCOR1 (Offset 0x03B0) ..... 513
19.4.134 P1OSDWSZ1 (Offset 0x03B4) ..... 514
19.4.135 P1OSDSSZ1 (Offset 0x03B8) ..... 514
19.4.136 P1OSDFSZ1 (Offset 0x03BC) ..... 514
19.4.137 P1OSDCOR2 (Offset 0x03C0) ..... 515
19.4.138 P1OSDWSZ2 (Offset 0x03C4) ..... 516
19.4.139 P1OSDSSZ2 (Offset 0x03C8) ..... 516
19.4.140 P1OSDFSZ2 (Offset 0x03CC) ..... 517
19.4.141 P1OSDCOR3 (Offset 0x03D0) ..... 517
19.4.142 P1OSDWSZ3 (Offset 0x03D4) ..... 518
19.4.143 P1OSDSSZ3 (Offset 0x03D8) ..... 519
19.4.144 P1OSDFSZ3 (Offset 0x03DC) ..... 519
19.4.145 INTSTS (Offset 0x03E0) ..... 520
19.4.146 INTMASK (Offset 0x03E4) ..... 521
19.5 Function Description ..... 523
19.5.1 Video Port ..... 523
19.5.2 De-interlacer ..... 526
19.5.3 Size-down ..... 526
19.5.4 OSD ..... 526
19.5.5 Color Space Conversion ..... 527
19.5.6 Window Clip ..... 527
19.5.7 VBI Extraction ..... 527
19.5.8 DMA ..... 527
19.5.9 YCbCr 4:2:2 Raster Order Sequence ..... 527
Chapter 20 Image Sensor Processor ..... 529
20.1 General Description ..... 530
20.2 Features ..... 530
20.3 Block Diagram ..... 531
20.4 Registers and Functions ..... 532
20.4.1 Summary of Control Registers ..... 532
20.4.2 ISP Global Registers ..... 538
20.5 Block Descriptions ..... 622
20.5.1 Sensor I/F ..... 622
20.5.2 Sensor Correction ..... 623

20.5.3 RGB Gain ..... 623
20.5.4 Resize ..... 624
20.6 Statistics of 3A, Histogram, and Motion Detection ..... 627
20.7 Register and Table Access ..... 627
Chapter 21 NAND-type Flash Controller ..... 629
21.1 General Description ..... 630
21.2 Features ..... 630
21.3 Block Diagram ..... 631
21.4 Summary of Control Registers ..... 632
21.5 Register Definitions ..... 636
21.5.1 ECC Status Registers 0 and 1 (Offset = 0x0000 and 0x0004) ..... 636
21.5.2 ECC Control Register (Offset = 0x0008) ..... 637
21.5.3 ECC Interrupt Enable Register (Offset = 0x0020) ..... 640
21.5.4 ECC Interrupt Status Register (Offset = 0x0024) ..... 641
21.5.5 ECC Status Clear Register (Offset = 0x0028) ..... 643
21.5.6 ECC Status of Spare Region Registers 0 and 1 (Offset = 0x002C and 0x0030) ..... 643
21.5.7 Device Busy/Ready Status Register (Offset = 0x0100) ..... 644
21.5.8 NANDC General Setting Register (Offset = 0x0104) ..... 645
21.5.9 Memory Attribute Setting Register (Offset = 0x0108) ..... 646
21.5.10 Spare Region Access Mode Register (Offset = 0x010C) ..... 647
21.5.11 Flash AC Timing Register 0 of Channel *x* (Offset = 0x0110 + *x**0x8, *x* = 0 ~ 7) ..... 647
21.5.12 Flash AC Timing Register 1 of Channel *x* (Offset = 0x0114 + *x**0x8, *x* = 0 ~ 7) ..... 649
21.5.13Flash AC Timing Register 2 of Channel *x* (Offset = 0x0190 + *x* * 0x8, *x* = 0 ~ 7) ..... 651
21.5.14 NANDC Interrupt Enable Register (Offset = 0x0150) ..... 658
21.5.15 NANDC Interrupt Status Register (Offset = 0x0154) ..... 658
21.5.16 NANDC Interrupt Status Register (Offset = 0x0154) ..... 659
21.5.17 Current Access Row Address Register (Offset = 0x0158 ~ 0x0174) ..... 659
21.5.18 Read Status Registers, 0 and 1 (Offset = 0x0178 ~ 0x017C) ..... 660
21.5.19 Address Toggle Bit Location Register (Offset = 0x0180) ..... 660
21.5.20 NANDC Software Reset Register (Offset = 0x0184) ..... 660
21.5.21 NANDC Auto-compare Pattern Register (Offset = 0x0188) ..... 661

21.5.22 Command Queue Status Register (Offset = 0x0200) ..... 662
21.5.23 Command Queue Flush Register (Offset = 0x0204)..... 662
21.5.24 Command Complete Counter (Offset = 0x0208) ..... 663
21.5.25 Command Complete Counter Reset (Offset = 0x020C) ..... 663
21.5.26 Command Format ..... 664
21.5.27 General Command Queue (Offset = 0x0280 ~ 0x0294) ..... 664
21.5.28 Command Queue (Offset = 0x0300 + *n* * 20 ~ 0x0314 + *n* * 0x20, *n* = 0 ~ 7)664
21.5.29 BMC Region Status Register (Offset = 0x0400) ..... 671
21.5.30 Region *n* User Mode Pointer Adjustment Register (Offset = 0x0404 + *n* * 0x4, *n* = 0 ~ 7)..... 672
21.5.31 DMA Mode Write Data Fill/Read Data Pop Register (Offset = 0x0424).... 672
21.5.32 Region Software Reset Register (Offset = 0x0428)..... 673
21.5.33 Force Region Fill Read Data Register (Offset = 0x042C)..... 673
21.5.34 Region *x* Remaining Sector Count of Read Data (Offset = 0x0430 + *x* * 0x4, *x* = 0 ~ 7)..... 674
21.5.35 Revision Number Register (Offset = 0x0500) ..... 674
21.5.36 Feature1 Register (Offset = 0x0504)..... 674
21.5.37 AHB Slave Memory Space Range Register (Offset = 0x0508)..... 676
21.5.38 Global Software Reset Register (Offset = 0x050C) ..... 677
21.5.39 AHB Data Slave Reset Register (Offset = 0x0510) ..... 678
21.5.40 ECC Correction Capability Register (Offset = 0x0514)..... 679
21.5.41 ECC Correction Capability Register (Offset = 0x0518)..... 680
21.5.42 Programmable Flow Control Register (Offset = 0x0600+*n* * 0x4, *n* = 0 ~ 31)680
21.5.43 Programmable OP-code Register (Offset = 0x0700 ~ 0x0704)..... 680
21.5.44 Spare SRAM Access Register (Offset = 0x1000 ~ 0x2FFF)..... 681
21.5.45 Data SRAM Access Register (Offset = 0x1_0000 ~ 0x1_1FFF)..... 681
21.6 AHB Data Slave Port..... 681
21.7 Buffer Management Controller ..... 682
21.8 AHB Register Slave Port..... 684
21.9 ECC Correction Error Handling..... 684
21.10 Auto Compare Error Handling..... 685
21.11 Abort Sequence..... 685
21.12 Rom/SRAM Behavior ..... 685
21.13 Host Controller Data Format ..... 686
21.14 Description of NANDC MicroCode ..... 688

21.14.1 MicroCode Description ..... 689
21.14.2 Command Register Setting ..... 693
21.14.3 Flow Rules ..... 695
21.14.4 Spare SRAM Organization ..... 698
21.14.5 User Data Position at Spare SRAM ..... 699
21.14.6 User Data, CRC, and ECC Usages ..... 702
21.14.7 Fixed Flow Command and Usage ..... 703
Chapter 22 Scaler ..... 715
22.1 General Description ..... 716
22.2 Features ..... 716
22.3 Block Diagram ..... 717
22.4 Modes of Operation ..... 717
22.5 Programming Model ..... 719
22.5.1 Mapping of Usual Special-purpose Registers ..... 719
22.5.2 Special-purpose Registers and Functions ..... 721
22.5.3 MDEST2 (Offset: 0x0260) ..... 743
22.6 Function Descriptions ..... 760
22.6.1 Video Port ..... 760
22.6.2 Size-down ..... 761
22.6.3 OSD ..... 761
22.6.4 Color Space Conversion ..... 762
22.6.5 Window Clip ..... 762
22.6.6 DMA ..... 762
Chapter 23 UART Controller ..... 769
23.1 General Description ..... 770
23.2 Features ..... 770
23.3 Block Diagram ..... 771
23.4 Programming Model ..... 772
23.4.1 Summary of IrDA Communications Controller Registers ..... 772
23.4.2 Register Descriptions ..... 772
Chapter 24 Synchronous Serial Port Controller ($I^2S$) ..... 787
24.1 General Description ..... 788
24.2 Features ..... 788
24.3 Block Diagram ..... 789

24.4 Programming Model ..... 789
24.4.1 SSP Control Registers Specification ..... 789
24.4.2 Register Descriptions ..... 790
Chapter 25 Secure Digital Memory Host Controller (SDC) ..... 797
25.1 General Description ..... 798
25.2 Features ..... 798
25.3 Block Diagram ..... 799
25.4 Programming Model ..... 800
25.4.1 Summary of SD Host Controller Registers ..... 800
25.4.2 Register Description ..... 801
25.4.3 Virtual Memory Register ..... 813
Chapter 26 $I^2C$ Bus Interface Controller ..... 815
26.1 General Description ..... 816
26.2 Features ..... 816
26.3 Block Diagram ..... 817
26.4 Programming Model ..... 817
26.4.1 Summary of FTIIC010 $I^2C$ Controller Registers ..... 817
26.4.2 Register Descriptions ..... 818
Chapter 27 GPIO Controller ..... 829
27.1 General Description ..... 830
27.2 Features ..... 830
27.3 Block Diagram ..... 831
27.4 Programming Model ..... 832
27.4.1 Summary of General Purpose I/O Registers ..... 832
27.4.2 Register Descriptions ..... 834
Chapter 28 Interrupt Controller ..... 841
28.1 General Description ..... 842
28.2 Features ..... 842
28.3 Block Diagram ..... 843
28.4 Interrupt Routing Table ..... 843
28.5 Programming Model ..... 845
28.5.1 Summary of Interrupt Controller Registers ..... 845
28.5.2 Register Descriptions ..... 846
Chapter 29 Timer ..... 857

29.1 General Description ..... 858
29.2 Features ..... 858
29.3 Block Diagram ..... 858
29.4 Programming Model ..... 859
29.4.1 Summary of Timer Registers ..... 859
29.4.2 Register Description ..... 860
Chapter 30 Timer With PWM Function ..... 867
30.1 General Description ..... 868
30.2 Features ..... 868
30.3 Block Diagram ..... 869
30.4 Programming Model ..... 870
30.4.1 Summary of Timer Registers ..... 870
30.4.2 Register Description ..... 871
30.4.3 Timer Control Register: TMX_CTRL (X = 1, 2, …, 8) ..... 873
30.4.4 Timer Counting Value Buffer Register: TMX_CNTB (X=1, 2, …, 8) ..... 874
30.4.5 Timer Counter Compare Buffer Register: TMX_CMPB (X = 1, 2, …, 8) ..... 874
30.4.6 Timer Count Observation Register: TMX_CNTO (X = 1, 2, …, 8) ..... 874
30.4.7 Revision Register: TMR_REV ..... 875
30.5 Basic Timer Operation ..... 875
30.5.1 Auto-reload and Double Buffering ..... 876
30.5.2 PWM Function ..... 877
30.5.3 Dead-zone Generation ..... 878
30.5.4 DMA Request Mode ..... 879
30.5.5 Initialization/Application Information ..... 880
Chapter 31 Watchdog Timer ..... 881
31.1 General Description ..... 882
31.2 Features ..... 882
31.3 Block Diagram ..... 882
31.4 Programming Model ..... 883
31.4.1 Summary of the WatchDog Timer Register ..... 883
31.4.2 Register Description ..... 883
31.4.3 WdRestart (Offset = 0x08) ..... 884
Chapter 32 Real Time Clock (RTC) ..... 887
32.1 General Description ..... 888

32.2 Features ..... 888
32.3 Block Diagram ..... 889
32.4 Block Description ..... 889
32.4.1 APB Interface ..... 889
32.4.2 RTC Alarm Registers ..... 890
32.4.3 RTC Control Register ..... 890
32.4.4 RTC Record Register ..... 890
32.4.5 Sync. Block ..... 890
32.4.6 RTC Counters ..... 891
32.4.7 RTC Auto-alarm Logic ..... 891
32.4.8 RTC Compare Logic ..... 891
32.4.9 Frequency Divider ..... 892
32.5 Programming Model ..... 892
32.5.1 Summary of Real Time Clock Registers ..... 892
32.5.2 Register Descriptions ..... 893
Chapter 33 DC Characteristics ..... 901
33.1 Absolute Maximum Ratings ..... 902
33.2 Recommended Operating Conditions ..... 902
33.3 I/O Pad Capacitance ..... 902
33.4 DC Characteristics of 3.3 V I/O Cells ..... 903

# LIST OF TABLES


Table 2-1. TFBGA-256 Signal Descriptions ..... 26
Table 2-2. E-PAD TQFP-176 Signal Descriptions ..... 31
Table 2-3. Jumper Settings ..... 35
Table 4-1. Relationship between FCLK/HCLK/DDR2CLK and PLL1 ..... 42
Table 4-2. Jumper Setting for Frequency Output of PLL1 ..... 43
Table 4-3. Summary of PMU Registers ..... 48
Table 4-4. ID Number Register (IDNMBR) ..... 49
Table 4-5. Jump Setting ..... 50
Table 4-6. OSC Control Register (OSCCR) ..... 51
Table 4-7. Power Mode Register (PMR) ..... 51
Table 4-8. Power Manager Control Register (PMCR) ..... 52
Table 4-9. Power Manager Status Register (PMSR) ..... 52
Table 4-10. Clock Source Control Register (CSCR) ..... 54
Table 4-11. PLL1 Control Register (PLL1CR) ..... 56
Table 4-12. PLL2 Control Register (PLL2CR) ..... 57
Table 4-13. AHB Module Clock Off Control Register (AHBMCLKOFF) ..... 57
Table 4-14. APB Module Clock Off Control Register (APBMCLKOFF) ..... 59
Table 4-15. Driving Capability and Slew Rate Control Register 0 (DCSRCR0) ..... 61
Table 4-16. Driving Capability and Slew Rate Control Register 1 (DCSRCR1) ..... 64
Table 4-17. Multi-function Port Setting Register 0 (MFPSR0) ..... 66
Table 4-18. Multi-function Port Setting Register 1 (MFPSR1) ..... 69
Table 4-19. Multi-function Port Setting Register 2 (MFPSR2) ..... 72
Table 4-20. Multi-function Port Setting Register 3 (MFPSR3) ..... 75
Table 4-21. Multi-function Port Setting Register 4 (MFPSR4) ..... 78
Table 4-22. Multi-function Port Setting Register 5 (MFPSR5) ..... 81
Table 4-23. System Controller Control Register (SCCR) ..... 84
Table 4-24. Divider Setting Register 0 (DSR0) ..... 86
Table 4-25. Divider Setting Register 1 (DSR1) ..... 86
Table 4-26. Divider Setting Register 2 (DSR2) ..... 87

Table 4-27. TVE DAC Control/Divider Setting Register 3 (DSR3) ..... 88
Table 4-28. DDR2 PHY Control Register (DPCR) ..... 89
Table 4-29. MAC PHY Control Register (MPCR) ..... 89
Table 4-30. MAC PHY ID Register (MPIDR) ..... 90
Table 4-31. Audio Codec Control Register 0 (ACR0) ..... 90
Table 4-32. Audio Codec ADC Control Register 1 (ACR1) ..... 92
Table 4-33. Audio Codec Control Register 2 (ACR2) ..... 95
Table 4-34. Audio Codec Control Register 3 (ACR3) ..... 96
Table 4-35. Audio Codec Control Register 4 (ACR4) ..... 98
Table 4-36. OTG Control Register 1 (OTGCR) ..... 100
Table 4-37. Pin-mux Table 1 ..... 101
Table 4-38. Pin-mux Table 2 ..... 101
Table 4-39. Pin-mux Table 3 ..... 101
Table 4-40. Pin-mux Table 4 ..... 102
Table 4-41. Pin-mux Table 4 ..... 102
Table 4-42. Pin-mux Table 5 ..... 103
Table 4-43. Pin-mux Table 6 ..... 103
Table 4-44. Pin-mux Table 7 ..... 103
Table 4-45. Pin-mux Table 8 ..... 104
Table 4-46. Pin-mux Table 9 ..... 104
Table 4-47. Pin-mux Table 10 ..... 105
Table 4-48. Pin-mux Table 11 ..... 105
Table 6-1. Summary of Control Registers ..... 115
Table 6-2. AHB Slave n Base/Size Register ..... 116
Table 6-3. Priority Control Register ..... 117
Table 6-4. Idle Count Register ..... 117
Table 6-5. Control Register ..... 117
Table 6-6. AHB Bus Request Enable Register ..... 119
Table 7-1. APB DMA REQ/ACK Mapping Table ..... 123
Table 7-2. Summary of APB Bridge Registers ..... 124
Table 7-3. Base/Size Register of APB Slave n ..... 125
Table 7-4. Source Addresses of DMA Channel Register ..... 126
Table 7-5. Destination Addresses of DMA Channel Register ..... 126
Table 7-6. Cycles of DMA Channels Register ..... 126

Table 7-7. Commands of DMA Channels Register ..... 127
Table 7-8. APB Control Register (0xc0) ..... 129
Table 7-9. APB Status Register (0xc4) ..... 130
Table 7-10. Revision Register ..... 130
Table 8-1. Summary of Static Memory Controller Register ..... 136
Table 8-2. Memory Bank Configuration Register (Offset = n * 8) ..... 137
Table 8-3. Memory Bank Configuration Register (Offset = (n * 8) + 4) ..... 139
Table 8-4. AST Register Setting Mapping ..... 140
Table 8-5. CTW Register Setting Mapping ..... 141
Table 8-6. AT1 Register Setting Mapping ..... 141
Table 8-7. WTC Register Setting Mapping ..... 141
Table 8-8. AHT Register Setting Mapping ..... 141
Table 8-9. TRNA Register Setting Mapping ..... 141
Table 8-10. AST Register Setting Mapping ..... 142
Table 8-11. CTW Register Setting Mapping ..... 142
Table 8-12. AT1 Register Setting Mapping ..... 142
Table 8-13. WTC Register Setting Mapping ..... 143
Table 8-14. AHT Register Setting Mapping ..... 143
Table 8-15. TRNA Register Setting Mapping ..... 143
Table 8-16. Shadow Status Register ..... 144
Table 9-1. Summary of DDR2 Controller Registers ..... 156
Table 9-2. Memory Controller Configuration Register (Offset 0x00) ..... 157
Table 9-3. Memory Controller State Control Register ..... 159
Table 9-4. Mode Register and Extended Mode Register Value Register ..... 161
Table 9-5. Extended Mode Register 2 and Extended Mode Register 3 Value Register ..... 161
Table 9-6. External Rank0/1 Register (Offset = 0x10) ..... 162
Table 9-7. Timing Parameter 0 Register (Offset = 0x14) ..... 164
Table 9-8. Timing Parameter 1 Register (Offset = 0x18) ..... 165
Table 9-9. Timing Parameter 2 Register (Offset = 0x1C) ..... 167
Table 9-10. DDR2 PHY Command and Data Block Control Register (Offset = 0x20) ..... 169
Table 9-11. DDR PHY Read Path DLL Delay Tuning Register (Offset = 0x24) ..... 172
Table 9-12. COMPBLK Control Register (Offset = 0x28) ..... 173
Table 9-13. Automatic Power-down Control Register (Offset = 0x2C) ..... 173
Table 9-14. Channel Arbitration Setup Register (Offset = 0x30) ..... 174

Table 9-15. Channel Arbiter Grant Count Register – A (Offset = 0x34) ..... 175
Table 9-16. Channel Arbiter Grant Count Register – B (Offset = 0x38) ..... 176
Table 9-17. Command Flush Control Register (Offset = 0x40) ..... 177
Table 9-18. Command Flush Status Register (Offset = 0x44) ..... 178
Table 9-19. AHB SPLIT Control Register (Offset = 0x48) ..... 178
Table 9-20. AHB INCR Read Pre-fetch Length 1 (Offset = 0xA0) ..... 180
Table 9-21. AHB INCR Read Pre-fetch Length 2 (Offset = 0xA4) ..... 181
Table 9-22. Initialization of Waiting Cycle Count 1 (Offset = 0xA8) ..... 182
Table 9-23. MA Table (AMTSEL = 2'b00) in 8-bit Mode ..... 183
Table 9-24. MA Table (AMTSEL = 2'b00) in 16-bit Mode ..... 184
Table 9-25. MA Table (AMTSEL = 2'b00) in 32-bit Mode ..... 185
Table 9-26. MA Table (AMTSEL = 2'b00) in 64-bit Mode ..... 186
Table 9-27. MA Table (AMTSEL = 2'b01) in 8-bit Mode ..... 188
Table 9-28. MA Table (AMTSEL = 2'b01) in 16-bit Mode ..... 189
Table 9-29. MA Table (AMTSEL = 2'b01) in 32-bit Mode ..... 190
Table 9-30. MA Table (AMTSEL = 2'b01) in 64-bit Mode ..... 191
Table 9-31. MA Table (AMTSEL = 2'b10) in 16-bit Mode ..... 193
Table 9-32. MA Table (AMTSEL = 2'b10) in 32-bit Mode ..... 194
Table 9-33. MA Table (AMTSEL = 2'b10) in 64-bit Mode ..... 195
Table 10-1. Address Map for Linked List Descriptor (Base Address: Cn_LLP[31:2]) ..... 203
Table 10-2. Control Field Definition in Linked List Descriptor ..... 203
Table 10-3. Total Transfer Size Definition in Linked List Descriptor ..... 203
Table 10-4. AHB DMA REQ/ACK Mapping Table ..... 208
Table 10-5. Summary of DMA Controller Registers ..... 210
Table 10-6. Interrupt Status Register (INT) ..... 212
Table 10-7. Terminal Count Interrupt Status Register (INT_TC) ..... 213
Table 10-8. Terminal Count Interrupt Status Clear Register (INT_TC_CLR) ..... 214
Table 10-9. Error/Abort Interrupt Status Register (INT_ERR/ABT) ..... 215
Table 10-10. Error/Abort Interrupt Status Clear Register (INT_ERR/ABT_CLR) ..... 217
Table 10-11. Terminal Count Status Register (TC) ..... 218
Table 10-12. Error/Abort Status Register (ERR/ABT) ..... 219
Table 10-13. Channel Enable Status Register (CH_EN) ..... 221
Table 10-14. Channel Busy Status Register (CH_BUSY) ..... 222
Table 10-15. Main Configuration Status Register (CSR) ..... 223

Table 10-16. Synchronization Register (SYNC) ..... 223
Table 10-17. Channel Control Register (Cn_CSR) ..... 225
Table 10-18. Channel Configuration Register (Cn_CFG) ..... 230
Table 10-19. Channel Source Address Register (Cn_SrcAddr) ..... 232
Table 10-20. Channel Destination Address Register (Cn_DstAddr) ..... 232
Table 10-21. Linked List Descriptor Pointer (Cn_LLP) ..... 232
Table 10-22. Transfer Size Register (Cn_SIZE) ..... 233
Table 11-1. Block Size of Each Mode ..... 243
Table 11-2. Cipher Operation Cycle ..... 244
Table 11-3. Summary of Control Registers ..... 244
Table 11-4. Bit Assignments of Encryption Control Register ..... 245
Table 11-5. Bit Assignments of FIFO Status Register ..... 246
Table 11-6. Bit Assignments of Parity Error Register ..... 247
Table 11-7. Bit Assignment of Key N Register ..... 247
Table 11-8. Bit Assignment of Initial Vector N Register ..... 247
Table 11-9. Bit Assignment of DMA Source Address Register ..... 248
Table 11-10. Bit Assignment of DMA Destination Address Register ..... 248
Table 11-11. Bit Assignments of DMA Transfer Size Register ..... 248
Table 11-12. Bit Assignments of DMA Control Register ..... 249
Table 11-13. Bit Assignments of FIFO Threshold Register ..... 249
Table 11-14. Bit Assignments of Interrupt Enable Register ..... 250
Table 11-15. Bit Assignments of Interrupt Status Register ..... 250
Table 11-16. Bit Assignments of Masked Interrupt Status Register ..... 250
Table 11-17. Bit Assignments of Interrupt Clear Register ..... 251
Table 11-18. Bit Assignment of Revision Register ..... 251
Table 11-19. Bit Assignment of Feature Register ..... 251
Table 11-20. Bit Assignment of Last Initial Vector N Register ..... 252
Table 11-21. Byte Sequence of Initial Vector ..... 252
Table 11-22. AES-128 Key Stream of Byte Sequence ..... 253
Table 11-23. AES-192 Key Stream of Byte Sequence ..... 254
Table 11-24. AES-256 Key Stream of Byte Sequence ..... 254
Table 11-25. DES Key Stream of Byte Sequence ..... 255
Table 11-26. Triple-DES Key Stream of Byte Sequence ..... 255
Table12-1. Register Summary ..... 260

Table12-2. HC Capability Register ..... 261
Table12-3. HC Structural Parameters ..... 261
Table 12-4. HC Capability Parameters ..... 261
Table 12-5. HC USB Command Register ..... 262
Table 12-6. HC USB Status Register ..... 264
Table 12-7. HC USB Interrupt Enable Register ..... 265
Table 12-8. HC Frame Index Register ..... 265
Table 12-9. HC Periodic Frame List Base Address Register ..... 266
Table 12-10. HC Current Asynchronous List Address Register ..... 266
Table 12-11. HC Port Status and Control Register ..... 266
Table 12-12. HC Misc. Register ..... 269
Table 12-13. OTG Control Status Register ..... 271
Table 12-14. OTG Interrupt Status Register ..... 273
Table 12-15. OTG Interrupt Enable Register ..... 274
Table 12-16. Global HC/OTG/DEV Interrupt Status Register ..... 275
Table 12-17. Global Mask of HC/OTG/DEV Interrupt ..... 275
Table 12-18. Device Main Control Register ..... 276
Table 12-19. Device Address Register ..... 277
Table 12-20. Device Test Register ..... 278
Table 12-21. Device SOF Frame Number Register ..... 278
Table 12-22. Device SOF Mask Timer Register ..... 279
Table 12-23. PHY Test Mode Selector Register ..... 279
Table 12-24. Device Vendor-specific I/O Control Register ..... 280
Table 12-25. Device CX Configuration Status Register ..... 280
Table 12-26. Device CX Configuration and FIFO Empty Status Register ..... 280
Table 12-27. Device Idle Counter Register ..... 281
Table 12-28. Device Mask of Interrupt Group Register ..... 282
Table 12-29. Device Mask of Interrupt Source Group 0 Register ..... 283
Table 12-30. Device Mask of Interrupt Source Group 1 Register ..... 283
Table 12-31. Device Mask of Interrupt Source Group 2 Register ..... 285
Table 12-32. Device Interrupt Group Register ..... 286
Table 12-33. Device Interrupt Source Group 0 Register ..... 286
Table 12-34. Device Interrupt Source Group 1 Register ..... 288
Table 12-35. Device Interrupt Source Group 2 Register ..... 289

Table 12-36. Device Receive Zero-Length Data Packet Register ..... 291
Table 12-37. Device Transfer Zero-Length Data Packet Register ..... 292
Table 12-38. Device Isochronous Sequential Error/Abort Register ..... 292
Table 12-39. Device IN Endpoint x MaxPacketSize Register ..... 293
Table 12-40. Device OUT Endpoint x MaxPacketSize Register ..... 294
Table 12-41. Device Endpoint 1 ~ 4 Map Register ..... 295
Table 12-42. Device Endpoint 5 ~ 8 Map Register ..... 296
Table 12-43. Device FIFO Map Register ..... 297
Table 12-44. Device FIFO Configuration ..... 298
Table 12-45. Device FIFO x Instruction and Byte Count Register ..... 300
Table 12-46. Device DMA Target FIFO Number Register ..... 301
Table 12-47. Device DMA Controller Parameter Setting 1 Register ..... 301
Table 12-48. Device DMA Controller Parameter Setting 2 Register ..... 302
Table 12-49. Device DMA Controller Parameter Setting 3 Register ..... 303
Table 15-1. Signal Mapping for Raw RGB Modes and Panel Types ..... 327
Table 15-2. RGB444, RGB555, and RGB565 Memory Formats in 16 bpp Raw RGB Mode ..... 327
Table 15-3. Pixel Sequence in YCbCr422 Mode ..... 328
Table 15-4. Component Locations/Sequences in YCbCr420 Mode ..... 329
Table 15-5. Palette RAM Data Structure ..... 330
Table 15-6. Font Attribute Structure ..... 331
Table 15-7. Summary of Control Registers ..... 335
Table 15-8. LCD Function Enable Parameters ..... 337
Table 15-9. LCD Panel Pixel Parameters ..... 338
Table 15-10. LCD Interrupt Enable Mask Parameters ..... 340
Table 15-11. LCD Interrupt Status Clear Parameters ..... 341
Table 15-12. LCD Interrupt Status Parameters ..... 341
Table 15-13. Frame Buffer Parameters ..... 342
Table 15-14. Image0 Frame Base Address ..... 343
Table 15-15. Image1 Frame Base Address ..... 343
Table 15-16. Image1 Frame Base Address ..... 343
Table 15-17. Image0 Frame Base Address ..... 343
Table 15-18. PatGen Pattern Bar Distance Parameter ..... 344
Table 15-19. FIFO Threshold Control Parameter ..... 345
Table 15-20. LCD Horizontal Timing Control Parameter ..... 345

Table 15-21. LCD Vertical Timing Control Parameters ..... 346
Table 15-22. LCD Vertical Back Porch Parameter ..... 347
Table 15-23. LCD Polarity Control Parameters ..... 347
Table 15-24. LCD TV Parameter0 ..... 348
Table 15-25. TV Parameter 1 ..... 350
Table 15-26. TV Parameter 2 ..... 350
Table 15-27. TV Parameter 3 ..... 350
Table 15-28. TV Parameter 4 ..... 351
Table 15-29. TV Parameter 5 ..... 351
Table 15-30. TV Parameter 6 ..... 351
Table 15-31. TV Parameter 7 ..... 351
Table 15-32. TV Parameter 8 ..... 352
Table 15-33. TV Parameter 9 ..... 352
Table 15-34. TV Parameter 10 ..... 352
Table 15-35. Horizontal Input Resolution (Offset 0x1100) ..... 353
Table 15-36. Vertical Input Resolution (Offset 0x1104) ..... 353
Table 15-37. Horizontal Output Resolution (Offset 0x1108) ..... 353
Table 15-38. Vertical Output Resolution (Offset 0x110C) ..... 354
Table 15-39. Scaler Control Register (Offset 0x1110) ..... 354
Table 15-40. Horizontal High Threshold Register (Offset 0x1114) ..... 355
Table 15-41. Horizontal Low Threshold Register (Offset 0x1118) ..... 355
Table 15-42. Vertical High Threshold Register (Offset 0x111C) ..... 356
Table 15-43. Vertical Low Threshold Register (Offset 0x1120) ..... 356
Table 15-44. Scaler Resolution Parameters (Offset 0x112C) ..... 357
Table 15-45. Virtual Screen Control Register (Offset 0x1500) ..... 358
Table 15-46. OSD Scaling and Dimension Control (Offset 0x2000) ..... 358
Table 15-47. OSD Position Control (Offset 0x2004) ..... 359
Table 15-48. OSD Foreground Color Control (Offset 0x2008) ..... 359
Table 15-49. OSD Background Color Control (Offset 0x200C) ..... 360
Table 15-50. OSD Font Database Write Accessing Port (Offset 0x8000 ~ 0xBFFC) ..... 361
Table 15-51. OSD Window Attribute Write Accessing Port (Offset 0xC000 ~ 0xC7FC) ..... 362
Table 16-1. Interrupt Status Register, ISR, 32'h0 (Offset: 0x00 ~ 0x03) ..... 365
Table 16-2. Interrupt Enable Register, IME, 32'h0 (Offset: 0x04 ~ 0x07) ..... 366
Table 16-3. MAC Most Significant Address Register, MAC_MADR, 32'h0 (Offset: 0x08 ~ 0x0B) ..... 366

Table 16-4. MAC Least Significant Address Register, MAC_LADR, 32'h0 (Offset: 0x0C ~ 0x0F)..... 366
Table 16-5. Multicast Address Hash Table 0 Register, MAHT0, 32'h0 (Offset: 0x10 ~ 0x13)..... 367
Table 16-6. Multicast Address Hash Table 1 Register, MAHT1, 32'h0 (Offset: 0x14 ~ 0x17)..... 367
Table 16-7. Transmit Poll Demand Register, TXPD, 32'h0 (Offset: 0x18 ~ 0x1B) ..... 367
Table 16-8. Receive Poll Demand Register, RXPD, 32'h0 (Offset: 0x1C ~ 0x1F) ..... 367
Table 16-9. Transmit Ring Base Address Register, TXR_BADR, 32'h0 (Offset: 0x20 ~ 0x23)..... 368
Table 16-10. Receive Ring Base Address Register, RXR_BADR, 32'h0 (Offset: 0x24 ~ 0x27) ..... 368
Table 16-11. Interrupt Timer Control Register, ITC, 32'h0 (Offset: 0x28 ~ 0x2B) ..... 368
Table 16-12. Automatic Polling Timer Control Register, APTC, 32'h0, (Offset: 0x2C ~ 0x2F)..... 372
Table 16-13. DMA Burst Length and Arbitration Control Register, DBLAC, 32'h0, (Offset: 0x30 ~ 0x33)373
Table 16-14. Revision Register, REVR, 32'h0, (Offset: 0x34 ~ 0x37) ..... 375
Table 16-15. Feature Register, FEAR, 32'h0, (Offset: 0x38 ~ 0x3B)..... 375
Table 16-16. MAC Control Register, MACCR, 32'h0, (Offset: 0x88 ~ 0x8B)..... 376
Table 16-17. MAC Status Register, MACSR, 32'h0, (Offset: 0x8C~ 0x8F) ..... 377
Table 16-18. PHY Control Register, PHYCR, 32'h0, (Offset: 0x90 ~ 0x93) ..... 377
Table 16-19. PHY Write Data Register, PHYWDATA, 32'h0, (Offset: 0x94 ~ 0x97) ..... 378
Table 16-20. Flow Control Register, FCR, 32'h0000_A400, (Offset: 0x98 ~ 0x9B) ..... 378
Table 16-21. Back Pressure Register, BPR, 32'h0000_0400, (Offset: 0x9C ~ 0x9F) ..... 379
Table 16-22. Wake-On-LAN Control Register, WOLCR, 32'h0, (Offset: 0xA0 ~ 0xA3)..... 380
Table 16-23. Wake-On-LAN Status Register, WOLSR, 32'h0, (Offset: 0xA4 ~ 0xA7) ..... 381
Table 16-24. Wake-up Frame CRC Register, WFCRC, 32'h0, (Offset: 0xA8 ~ 0xAB)..... 381
Table 16-25. Wake-up Frame Byte Mask 1st Double Word Register, WFBM1, 32'h0, (Offset: 0xB0 ~ 0xB3) ..... 382
Table 16-26. Wake-up Frame Byte Mask 2nd Double Word Register, WFBM2, 32'h0, (Offset: 0xB4 ~ 0xB7) ..... 383
Table 16-27. Wake-up Frame Byte Mask 3rd Double Word Register, WFBM3, 32'h0, (Offset: 0xB8 ~ 0xBB)..... 384
Table 16-28. Wake-up Frame Byte Mask 4th Double Word Register, WFBM4, 32'h0, (Offset: 0xBC ~ 0xBF)..... 385
Table 16-29. Test Seed Register, TS, 32'h0, (Offset: 0xC4 ~ 0xC7)..... 385
Table 16-30. DMA/FIFO State Register, DMAFIFOS, 32'h0, (Offset: 0xC8 ~ 0xCB) ..... 385
Table 16-31. Test Mode Register, TM, 32'h0, (Offset: 0xCC ~ 0xCF)..... 386
Table 16-32. TX_MCOL and TX_SCOL Counter Register, 32'h0, (Offset: 0xD4 ~ 0xD7) ..... 387
Table 16-33. RPF and AEP Counter Register, 32'h0, (Offset: 0xD8 ~ 0xDB) ..... 387
Table 16-34. XM and PG Counter Register, 32'h0, (Offset: 0xDC ~ 0xDF)..... 387

Table 16-35. RUNT_CNT and TLCC Counter Register, 32'h0, (Offset: 0xE0 ~ 0xE3) ..... 388
Table 16-36. CRCER_CNT and FTL_CNT Counter Register, 32'h0, (Offset: 0xE4 ~ 0xE7) ..... 388
Table 16-37. RLC and RCC Counter Register, 32'h0, (Offset: 0xE8 ~ 0xEB) ..... 388
Table 16-38. BROC Counter Register, 32'h0, (Offset: 0xEC ~ 0xEF) ..... 388
Table 16-39. MULCA Counter Register, 32'h0, (Offset: 0xF0 ~ 0xF3) ..... 389
Table 16-40. RP Counter Register, 32'h0, (Offset: 0xF4 ~ 0xF7) ..... 389
Table 16-41. XP Counter Register, 32'h0, (Offset: 0xF8 ~ 0xFB) ..... 389
Table 16-42. Wake-up Frame Format ..... 401
Table 16-43. FTMAC110 Frame Structure and Frame Formats ..... 408
Table 16-44. PHY Bit Stream Format ..... 409
Table 17-1. FTTVE100_S Controller Register Summary ..... 415
Table 17-2. TVE Video Format Control Register ..... 416
Table 17-3. TVE Video Input Control Register ..... 417
Table 17-4. TVE Video Output Control Register ..... 418
Table 17-5. TVE Power-down Control Registers ..... 419
Table 17-6. TVE Software Reset Control Register ..... 419
Table 18-1. Control Register Summary ..... 423
Table 18-2. Forward Luma/Y Start address Register ..... 425
Table 18-3. Forward Luma/Y Start Address Register ..... 425
Table 18-4. Current Luma/Y Start Address Register ..... 425
Table 18-5. Next Luma/Y Start Address Register ..... 426
Table 18-6. Current Chroma/CbCr Start Address Register ..... 426
Table 18-7. Next Chroma/CbCr Start Address Register ..... 426
Table 18-8. Frame/Field Height Register ..... 426
Table 18-9. Frame Width Register ..... 427
Table 18-10. Command Register ..... 427
Table 18-11. Command Register ..... 428
Table 18-12. Destination Chroma/CbCr Start Address Register ..... 428
Table 18-13. Destination Chroma/CbCr Start Address Register ..... 428
Table 18-14. Motion Detection MS Mask Register for Low 32 Pixels ..... 429
Table 18-15. Motion Detection MS Mask Register for High 32 Pixels ..... 429
Table 18-16. Date and ID Information Register ..... 429
Table 18-17. Internal Control Status Register ..... 429
Table 18-18. Internal Control Status Register ..... 429

Table 18-19. De-noise Conifigure Register ..... 430
Table 18-20. De-noise Top Luma Variance Register0 ..... 430
Table 18-21. De-noise Top Luma Variance Register1 ..... 431
Table 18-22. De-noise Top Luma Variance Register2 ..... 431
Table 18-23. De-noise Top Luma Variance Register3 ..... 432
Table 18-24. De-noise Top Chroma(Cb) Variance Register0 ..... 432
Table 18-25. De-noise Top Chroma (Cb) Variance Register1 ..... 432
Table 18-26. De-noise Top Chroma (Cb) Variance Register2 ..... 433
Table 18-27. De-noise Top Chroma (Cb) Variance Register3 ..... 433
Table 18-28. De-noise Top Chroma (Cr) Variance Register0 ..... 433
Table 18-29. De-noise Top Chroma (Cr) Variance Register1 ..... 434
Table 18-30. De-noise Top Chroma (Cr) Variance Register2 ..... 434
Table 18-31. De-noise Top Chroma (Cr) Variance Register3 ..... 435
Table 19-1. Mapping of Usual Special-purpose Registers ..... 439
Table 19-2. VCAPEN Register ..... 444
Table 19-3. VCAPUPD Register ..... 445
Table 19-4. CAPCLK Register ..... 447
Table 19-5. PVSIZE0 Register ..... 447
Table 19-6. PVSIZE1 Register ..... 448
Table 19-7. PVSIZE2 Register ..... 448
Table 19-8. P1SHARPEN Register ..... 449
Table 19-9. P1SIZE0 Register ..... 449
Table 19-10. P1SIZE1 Register ..... 449
Table 19-11. P1SIZE2 Register ..... 450
Table 19-12. DFORMAT Register ..... 450
Table 19-13. GBATRB Register ..... 451
Table 19-14. FMRATE Register ..... 453
Table 19-15. SRCIF Register ..... 455
Table 19-16. SRCSIZE0 Register ..... 456
Table 19-17. P0SIZE3 Register ..... 456
Table 19-18. P0SIZE4 Register ..... 456
Table 19-19. P0SIZE5 Register ..... 457
Table 19-20. SRCSIZE1 Register ..... 457
Table 19-21. P1SIZE3 Register ..... 458

Table 19-22. P1SIZE4 Register ..... 458
Table 19-23. P1SIZE5 Register ..... 459
Table 19-24. P1SHARPEN Register ..... 459
Table 19-25. P0MSK0COR Register ..... 459
Table 19-26. P0MSK0SIZE0 Register ..... 460
Table 19-27. P0MSK0SIZE1 Register ..... 460
Table 19-28. P0MSK1COR Register ..... 462
Table 19-29. P0MSK1SIZE0 Register ..... 462
Table 19-30. P0MSK1SIZE1 Register ..... 463
Table 19-31. P0MSK2COR Register ..... 463
Table 19-32. P0MSK2SIZE0 Register ..... 464
Table 19-33. P0MSK2SIZE1 Register ..... 464
Table 19-34. P0MSK3COR Register ..... 464
Table 19-35. P0MSK3SIZE0 Register ..... 465
Table 19-36. P0MSK3SIZE1 Register ..... 465
Table 19-37. P0MSK4COR Register ..... 466
Table 19-38. P0MSK4SIZE0 Register ..... 466
Table 19-39. P0MSK4SIZE1 Register ..... 467
Table 19-40. P0MSK5COR Register ..... 467
Table 19-41. P0MSK0SIZE0 Register ..... 468
Table 19-42. P0MSK5SIZE1 Register ..... 468
Table 19-43. P0MSK6COR Register ..... 468
Table 19-44. P0MSK06SIZE0 Register ..... 469
Table 19-45. P0MSK6SIZE1 Register ..... 469
Table 19-46. P0MSK6COR Register ..... 470
Table 19-47. P0MSK7SIZE0 Register ..... 470
Table 19-48. P0MSK7SIZE1 Register ..... 471
Table 19-49. P1MSK0COR Register ..... 471
Table 19-50. P1MSK0SIZE0 Register ..... 472
Table 19-51. P1MSK0SIZE1 Register ..... 472
Table 19-52. P1MSK1COR Register ..... 472
Table 19-53. P1MSK1SIZE0 Register ..... 473
Table 19-54. P1MSK1SIZE1 Register ..... 473
Table 19-55. P1MSK2COR Register ..... 474

Table 19-56. P1MSK2SIZE0 Register ..... 474
Table 19-57. P1MSK2SIZE1 Register ..... 475
Table 19-58. P1MSK3COR Register ..... 475
Table 19-59. P1MSK3SIZE0 Register ..... 476
Table 19-60. P1MSK3SIZE1 Register ..... 476
Table 19-61. P1MSK4COR Register ..... 476
Table 19-62. P1MSK4SIZE0 Register ..... 477
Table 19-63. P1MSK4SIZE1 Register ..... 477
Table 19-64. P1MSK5COR Register ..... 478
Table 19-65. P1MSK0SIZE0 Register ..... 478
Table 19-66. P1MSK5SIZE1 Register ..... 479
Table 19-67. P1MSK6COR Register ..... 479
Table 19-68. P1MSK06SIZE0 Register ..... 480
Table 19-69. P1MSK6SIZE1 Register ..... 480
Table 19-70. P1MSK6COR Register ..... 480
Table 19-71. P1MSK7SIZE0 Register ..... 481
Table 19-72. P1MSK7SIZE1 Register ..... 481
Table 19-73. DICTRL0 Register ..... 481
Table 19-74. DICTRL1 Register ..... 482
Table 19-75. MISC0 Register ..... 482
Table 19-76. PDMA0 Register ..... 483
Table 19-77. P1DMA0 Register ..... 485
Table 19-78. PDMA1 Register ..... 486
Table 19-79. P1DMA1 Register ..... 486
Table 19-80. P0MDEST0 Register ..... 487
Table 19-81. P0MDEST1 Register ..... 487
Table 19-82. P0MDEST2 Register ..... 487
Table 19-83. P0MDEST3 Register ..... 487
Table 19-84. P0MDEST4 Register ..... 488
Table 19-85. P0MDEST5 Register ..... 488
Table 19-86. P1MDEST0 Register ..... 489
Table 19-87. P1MDEST1 Register ..... 489
Table 19-88. P1MDEST2 Register ..... 489
Table 19-89. P1MDEST3 Register ..... 489

Table 19-90. P1MDEST4 Register ..... 490
Table 19-91. P1MDEST5 Register ..... 490
Table 19-92. VBICTRL0 Register ..... 490
Table 19-93. VBICTRL1 Register ..... 490
Table 19-94. VBICTRL2 Register ..... 491
Table 19-95. VBICTRL3 Register ..... 491
Table 19-96. P0OSDFONT Register ..... 491
Table 19-97. P0OSDDISP Register ..... 492
Table 19-98. P0OSDREAD Register ..... 493
Table 19-99. P0OSDEN Register ..... 493
Table 19-100. P0OSDPAT0 Register ..... 494
Table 19-101. P0OSDPAT1 Register ..... 494
Table 19-102. P0OSDPAT2 Register ..... 494
Table 19-103. P0OSDPAT3 Register ..... 495
Table 19-104. P0OSDPAT4 Register ..... 495
Table 19-105. P0OSDPAT5 Register ..... 495
Table 19-106. P0OSDPAT6 Register ..... 496
Table 19-107. P0OSDCOR0 Register ..... 496
Table 19-108. P0OSDWSZ0 Register ..... 497
Table 19-109. P0OSDSSZ0 Register ..... 498
Table 19-110. P0OSDFSZ0 Register ..... 498
Table 19-111. P0OSDCOR1 Register ..... 499
Table 19-112. P0OSDWSZ1 Register ..... 500
Table 19-113. P0OSDSSZ1 Register ..... 501
Table 19-114. P0OSDFSZ1 Register ..... 501
Table 19-115. P0OSDCOR2 Register ..... 502
Table 19-116. P0OSDWSZ2 Register ..... 503
Table 19-117. P0OSDSSZ2 Register ..... 503
Table 19-118. P0OSDFSZ2 Register ..... 503
Table 19-119. P0OSDCOR3 Register ..... 504
Table 19-120. P0OSDWSZ3 Register ..... 505
Table 19-121. P0OSDSSZ3 Register ..... 506
Table 19-122. P0OSDFSZ3 Register ..... 506
Table 19-123. P1OSDFONT Register ..... 506

Table 19-124. P1OSDDISP Register ..... 507
Table 19-125. P1OSDREAD Register ..... 507
Table 19-126. P1OSDEN Register ..... 507
Table 19-127. P1OSDPAT0 Register ..... 508
Table 19-128. P1OSDPAT1 Register ..... 508
Table 19-129. P1OSDPAT2 Register ..... 509
Table 19-130. P1OSDPAT3 Register ..... 509
Table 19-131. P1OSDPAT4 Register ..... 509
Table 19-132. P1OSDPAT5 Register ..... 510
Table 19-133. P1OSDPAT6 Register ..... 510
Table 19-134. P1OSDCOR0 Register ..... 510
Table 19-135. P1OSDWSZ0 Register ..... 511
Table 19-136. P1OSDSSZ0 Register ..... 512
Table 19-137. P1OSDFSZ0 Register ..... 512
Table 19-138. P1OSDCOR1 Register ..... 513
Table 19-139. P1OSDWSZ1 Register ..... 514
Table 19-140. P1OSDSSZ1 Register ..... 514
Table 19-141. P1OSDFSZ1 Register ..... 514
Table 19-142. P1OSDCOR2 Register ..... 515
Table 19-143. P1OSDWSZ2 Register ..... 516
Table 19-144. P1OSDSSZ2 Register ..... 516
Table 19-145. P1OSDFSZ2 Register ..... 517
Table 19-146. P1OSDCOR3 Register ..... 517
Table 19-147. P1OSDWSZ3 Register ..... 518
Table 19-148. P1OSDSSZ3 Register ..... 519
Table 19-149. P1OSDFSZ3 Register ..... 519
Table 19-150. INTSTS Register ..... 520
Table 19-151. INTMASK Register ..... 521
Table 19-152. ITU-R BT. 656 EAV and SAV Sequence ..... 523
Table 20-1. ISP Terminology ..... 531
Table 20-2. Summary of Control Registers ..... 532
Table 20-3. ISP Function Enable Register ..... 538
Table 20-4. CMI and CU Register ..... 540
Table 20-5. SW Reset Register ..... 542

Table 20-6. CAP Style and Trigger Register ..... 542
Table 20-7. Command Buffer Control Register ..... 543
Table 20-8. Sensor Interface Register ..... 544
Table 20-9. Gamma Control Register ..... 545
Table 20-10. Sensor Size Register ..... 546
Table 20-11. Sensor Active Window start Point Register ..... 546
Table 20-12. Sensor Active Window Size Register ..... 547
Table 20-13. Sensor Blanking Register ..... 547
Table 20-14. Image Target Size Register ..... 547
Table 20-15. Target Crop Window Start Point Register ..... 548
Table 20-16. Target Crop Window Size Register ..... 548
Table 20-17. Destination Format Register ..... 548
Table 20-18. Memory Source Size Register ..... 549
Table 20-19. Memory Source Format Register ..... 549
Table 20-20. Optical Black Setting Register ..... 550
Table 20-21. Optical Black Threshold Register ..... 550
Table 20-22. Optical Black Offset Registers ..... 551
Table 20-23. Optical Black Read Register ..... 552
Table 20-24. Shading Correction Center Point Register ..... 552
Table 20-25. Shading Correction Shift Register ..... 552
Table 20-26. Shading Correction Offset Register ..... 553
Table 20-27. Raw Blending Coefficients Register ..... 553
Table 20-28. Dark Raw Precision Register ..... 553
Table 20-29. Defect Pixel Correction Control Register ..... 554
Table 20-30. Defect Pixel Correction Threshold Register ..... 554
Table 20-31. Crosstalk Threshold Register ..... 555
Table 20-32. Crosstalk Weighting Register ..... 555
Table 20-33. RB Pre-gain Register ..... 555
Table 20-34. G Pre-gain Register ..... 556
Table 20-35. RB Gain Register ..... 556
Table 20-36. G Gain Register ..... 556
Table 20-37. Color Correction Matrix 00 Register ..... 557
Table 20-38. Color Correction Matrix 01 Register ..... 557
Table 20-39. Color Correction Matrix 10 Register ..... 557

Table 20-40. Color Correction Matrix 11 Register ..... 558
Table 20-41. Color Correction Matrix 20 Register ..... 558
Table 20-42. Color Correction Matrix 21 Register ..... 558
Table 20-43. Color Space Conversion Matrix 00 Register ..... 559
Table 20-44. Color Space Conversion Matrix 01 Register ..... 559
Table 20-45. Color Space Conversion Matrix 10 Register ..... 560
Table 20-46. Color Space Conversion Matrix 11 Register ..... 560
Table 20-47. Color Space Conversion Matrix 20 Register ..... 560
Table 20-48. Color Space Conversion Matrix 21 Register ..... 561
Table 20-49. Contrast Enhancement Control 0 Register ..... 561
Table 20-50. Contrast Enhancement Control 1 Register ..... 561
Table 20-51. Saturation Control Register ..... 562
Table 20-52. Sharpness Control Register ..... 563
Table 20-53. FCS Control Register ..... 563
Table 20-54. HBLANK Extended Register ..... 564
Table 20-55. AE & AWB Control Register ..... 564
Table 20-56. AF & MD Control Register ..... 566
Table 20-57. HSTG Source Register ..... 567
Table 20-58. AE Window Start Point Register ..... 568
Table 20-59. AE Window Size Register ..... 568
Table 20-60. AE Window Number Register ..... 568
Table 20-61. AE Window Gap Register ..... 569
Table 20-62. AE STA Threshold Register ..... 569
Table 20-63. AWB Window Start Point Register ..... 570
Table 20-64. AWB Window Size Register ..... 570
Table 20-65. AWB Window Number Register ..... 571
Table 20-66. AWB Window Gap Register ..... 571
Table 20-67. AWB STA Threshold Register ..... 571
Table 20-68. AF Pre-Curve Point Register ..... 572
Table 20-69. AF Pre-Curve Slope 0 Register ..... 572
Table 20-70. AF Pre-Curve Slope 1 Register ..... 572
Table 20-71. AF Window Start Point Register ..... 573
Table 20-72. AF Window Size Register ..... 573
Table 20-73. AF Window Number Register ..... 573

Table 20-74. AF Window Gap Register ..... 574
Table 20-75. AF HF1 Threshold Register ..... 574
Table 20-76. AF HF2 Threshold Register ..... 574
Table 20-77. AF VF Threshold Register ..... 575
Table 20-78. AF HF1 Coefficient 01Register ..... 575
Table 20-79. AF HF1 Coefficient 23 Register ..... 576
Table 20-80. AF HF1 Coefficient 45 Register ..... 576
Table 20-81. AF HF1 Coefficient 67 Register ..... 576
Table 20-82. AF HF2 Coefficient 01Register ..... 577
Table 20-83. AF HF2 Coefficient 23 Register ..... 577
Table 20-84. AF HF2 Coefficient 45 Register ..... 577
Table 20-85. AF HF2 Coefficient 67 Register ..... 578
Table 20-86. AF VF Coefficient 01 Register ..... 578
Table 20-87. AF VF Coefficient 23 Register ..... 578
Table 20-88. AF VF Coefficient 4 Register ..... 579
Table 20-89. MD Window Start Point Register ..... 579
Table 20-90. MD Window Size Register ..... 579
Table 20-91. MD Window Number Register ..... 580
Table 20-92. MD Window Gap Register ..... 580
Table 20-93. MD Control 0 Register ..... 580
Table 20-94. MD Control 1 Register ..... 581
Table 20-95. MD Control 2 Register ..... 581
Table 20-96. MD Control 3 Register ..... 581
Table 20-97. HSTG Window Start Point Register ..... 582
Table 20-98. HSTG Window Size Register ..... 582
Table 20-99. DMA Control Register ..... 582
Table 20-100. DMA FIFO Water Mark Register ..... 584
Table 20-101. Source Memory Base Address Register ..... 585
Table 20-102. Source Memory Pitch Register ..... 585
Table 20-103. Destination Memory Base Address 0 Register ..... 585
Table 20-104. Destination Memory Base Address 1 Register ..... 585
Table 20-105. Destination Memory Pitch Register ..... 586
Table 20-106. Destination Memory Cb Base Address 0 Register ..... 586
Table 20-107. Destination Memory Cb Base Address 1 Register ..... 586

Table 20-108. Destination Memory Cr Base Address 0 Register ..... 586
Table 20-109. Destination Memory Cr Base Address 1 Register ..... 587
Table 20-110. Command 0 Base Address Register ..... 587
Table 20-111. Command X Base Address Register ..... 587
Table 20-112. Command 1 Base Address Register ..... 587
Table 20-113. AE Base Address 0 Register ..... 588
Table 20-114. AE Base Address 1 Register ..... 588
Table 20-115. AWB Base Address 0 Register ..... 588
Table 20-116. AWB Base Address 1 Register ..... 588
Table 20-117. AF Base Address 0 Register ..... 589
Table 20-118. AF Base Address 1 Register ..... 589
Table 20-119. HSTG Base Address Register ..... 589
Table 20-120. MD Base Address Register ..... 589
Table 20-121. Line Trigger Threshold Register ..... 590
Table 20-122. Sensor I/F Image Size Info Register ..... 590
Table 20-123. Size-down Image Size Info Register ..... 590
Table 20-124. Frame Counter Register ..... 591
Table 20-125. User Defined Register ..... 591
Table 20-126. STA Ready Status Register ..... 591
Table 20-127. ISP Status Register ..... 595
Table 20-128. ISP Interrupt Register ..... 597
Table 20-129. ISP Interrupt Mask Register ..... 599
Table 20-130. ISP State Machine 0 Register ..... 601
Table 20-131. ISP State Machine 1 Register ..... 601
Table 20-132. ISP Test Pattern Register ..... 602
Table 20-133. RDN Filter Threshold Register ..... 604
Table 20-134. RDN Filter Weighting Register ..... 604
Table 20-135. CDN Filter Threshold Register ..... 604
Table 20-136. CDN Filter Weighting Register ..... 605
Table 20-137. SP LPF Control Threshold Register ..... 605
Table 20-138. SHDC RB Up Kx Register ..... 605
Table 20-139. SHDC G Up Kx Register ..... 606
Table 20-140. SHDC RB Down Kx Register ..... 606
Table 20-141. SHDC G Down Kx Register ..... 607

Table 20-142. SHDC RB Right Kx Register ..... 607
Table 20-143. SHDC G Right Kx Register ..... 607
Table 20-144. SHDC RB Left Kx Register ..... 608
Table 20-145. SHDC G Left Kx Register ..... 608
Table 20-146. SP Gain 0 Register ..... 609
Table 20-147. SP Gain 1 Register ..... 609
Table 20-148. SP Gain 2 Register ..... 609
Table 20-149. SP Gain 3 Register ..... 610
Table 20-150. SP Gain 4 Register ..... 610
Table 20-151. SP Weighting 0 Register ..... 610
Table 20-152. SP Weighting N Register ..... 611
Table 20-153. Gamma R LUT 0 Register ..... 611
Table 20-154. Gamma R LUT N1 Register ..... 612
Table 20-155. Gamma R LUT 2048 Register ..... 612
Table 20-156. Gamma R LUT N2 Register ..... 612
Table 20-157. Gamma R LUT 3072 Register ..... 613
Table 20-158. Gamma R LUT N3 Register ..... 613
Table 20-159. Gamma G LUT Register ..... 613
Table 20-160. Gamma B LUT Register ..... 614
Table 20-161. Raw R Mapping Segment 0 Register ..... 614
Table 20-162. Raw R Mapping Segment 1 Register ..... 614
Table 20-163. Raw R Mapping Segment N Register ..... 615
Table 20-164. Raw B Mapping Segment N Register ..... 615
Table 20-165. Raw Gr Mapping Segment N Register ..... 615
Table 20-166. Raw Gb Mapping Segment N Register ..... 616
Table 20-167. YCC Luma Curve Segment N Register ..... 616
Table 20-168. YCC Chroma Curve Segment N Register ..... 616
Table 20-169. CS Gain Curve Segment N Register ..... 617
Table 20-170. CS Weighting Segment N Register ..... 617
Table 20-171. BCS Gain Curve Segment N Register ..... 618
Table 20-172. IE Saturation Gain Segment N Register ..... 618
Table 20-173. IE Saturation Hue Segment N Register ..... 619
Table 20-174. IE Saturation to Y Offset Segment N Register ..... 620
Table 20-175. Hue Rotation θ Table Register ..... 620

Table 20-176. CE Fine Tuning Gain Register ..... 621
Table 21-1. Summary of Control Registers ..... 632
Table 21-2. ECC Status Register 0 (Offset = 0x00) ..... 636
Table 21-3. ECC Status Register 1 (Offset = 0x04) ..... 636
Table 21-4. ECC Control Register (Offset = 0x0008) ..... 637
Table 21-5. Threshold Number of ECC Error Bits Register 0 (Offset = 0x0010) ..... 637
Table 21-6. Threshold Number of ECC Error Bits Register 1 (Offset = 0x0014) ..... 638
Table 21-7. Permitted Number of ECC Correction Capability Bits ..... 639
Table 21-8. Number of ECC Correction Capability Bits Register 0 (Offset = 0x0018) ..... 639
Table 21-9. Number of ECC Correction Capability Bits Register 1 (Offset = 0x001C) ..... 640
Table 21-10. ECC Interrupt Enable Register (Offset = 0x0020) ..... 640
Table 21-11. ECC Interrupt Enable Register (Offset = 0x0024) ..... 641
Table 21-12. ECC Status Clear Register (Offset = 0x0024) ..... 643
Table 21-13. ECC Status of Spare Region Register 0 (Offset = 0x2C) ..... 643
Table 21-14. ECC Status of Spare Region Register 1 (Offset = 0x30) ..... 644
Table 21-15. Device Busy/Ready Status Register (Offset = 0x0100) ..... 644
Table 21-16. NANDC General Setting Register (Offset = 0x0104) ..... 645
Table 21-17. Memory Attribute Setting Register (Offset = 0x0108) ..... 646
Table 21-18. Spare Region Access Mode Register (Offset = 0x010C) ..... 647
Table 21-19. Flash AC Timing Register 0 (Offset = 0x0110 + x*0x8, x = 0 ~ 7) ..... 648
Table 21-20. Flash AC Timing Register 1 (Offset = 0x0114 + x*0x8, x = 0 ~ 7) ..... 649
Table 21-21. Flash AC Timing Register 2 (Offset = 0x0190 + x*0x8, x = 0 ~ 7) ..... 651
Table 21-22. NANDC Interrupt Enable Register (Offset = 0x0150) ..... 658
Table 21-23. NANDC Interrupt Status Register (Offset = 0x0154) ..... 658
Table 21-24. NANDC Interrupt Status Register (Offset = 0x0154) ..... 659
Table 21-25. Current Access Row Address Register (Offset = 0x0158 ~ 0x0174) ..... 659
Table 21-26. Read Status Register 0 (Offset = 0x0178) ..... 660
Table 21-27. Read Status Register 1 (Offset = 0x017C) ..... 660
Table 21-28. Address Toggle Bit Location Register (Offset = 0x0180) ..... 660
Table 21-29. NANDC Software Reset Register (Offset = 0x0184) ..... 661
Table 21-30. NANDC Auto-compare Pattern Register (Offset = 0x0188) ..... 661
Table 21-31. Command Queue Status Register (Offset = 0x0200) ..... 662
Table 21-32. Command Queue Flush Register (Offset = 0x0204) ..... 662
Table 21-33. Command Queue Flush Register (Offset = 0x0208) ..... 663

Table 21-34. Command Queue Flush Register (Offset = 0x020C) ..... 663
Table 21-35. Command Queue First Word (Offset = 0x0300 + n * 0x20, n = 0 ~ 7) ..... 664
Table 21-36. Command Queue Second Word (Offset = 0x0304 + n * 0x20, n = 0 ~ 7) ..... 666
Table 21-37. Command Queue Third Word (Offset = 0x0308 + n * 0x20, n = 0 ~ 7) ..... 666
Table 21-38. Command Queue Fourth Word (Offset = 0x030C + n * 0x20, n = 0 ~ 7) ..... 666
Table 21-39. Command Queue Fifth Word (Offset = 0x0310 + n * 0x20, n = 0 ~ 7) ..... 667
Table 21-40. Command Queue Sixth Word (Offset = 0x0314 + n * 0x20, n = 0 ~ 7) ..... 667
Table 21-41. Spare Number Usage ..... 670
Table 21-42. BMC Region Status Register (Offset = 0x0400) ..... 671
Table 21-43. Region n User Mode Pointer Adjustment Register (Offset = 0x0404 + n * 0x4, n = 0 ~ 7)672
Table 21-44. DMA Mode Write Data Fill/Read Data Pop Register (Offset = 0x0424) ..... 673
Table 21-45. Region Software Reset Register (Offset = 0x0428) ..... 673
Table 21-46. Force Region Fill Read Data Register (Offset = 0x042C) ..... 674
Table 21-47. Region x Remaining Sector Count of Read Data (Offset = 0x0430 + x * 0x4, x = 0 ~ 7) . 674
Table 21-48. Revision Number Register (Offset = 0x0500) ..... 674
Table 21-49. Feature Register (Offset = 0x0504) ..... 674
Table 21-50. AHB Slave Memory Space Range Register (Offset = 0x0508) ..... 676
Table 21-51. Global Software Reset Register (Offset = 0x050C) ..... 678
Table 21-52. Global Software Reset Register (Offset = 0x0510) ..... 678
Table 21-53. Feature Register (Offset = 0x0514) ..... 679
Table 21-54. dqs_in_delay Register (Offset = 0x0518) ..... 680
Table 21-55. Programmable Flow Control Register (Offset = 0x0600+n*0x4, n = 0 ~ 31) ..... 680
Table 21-56. Programmable OP-code Register (Offset = 0x0700) ..... 680
Table 21-57. Programmable OP-code Register (Offset = 0x0704) ..... 681
Table 21-58. Summary of MicroCode ..... 688
Table 21-59. opcode ..... 689
Table 21-60. Command Register Setting ..... 693
Table 22-1. Mapping of Usual Special-purpose Registers ..... 719
Table 22-2. SCAREN Register ..... 721
Table 22-3. SCARUPD Register ..... 722
Table 22-4. SCARCLK Register ..... 723
Table 22-5. ASPMJ Register ..... 724
Table 22-6. SIZE0 Register ..... 724
Table 22-7. SIZE1 Register ..... 725

Table 22-8. SIZE2 Register ..... 725
Table 22-9. BORDER_SIZE Register ..... 725
Table 22-10. BORDER_COLOR Register ..... 726
Table 22-11. DFORMAT Register ..... 727
Table 22-12. SRCSEL Register ..... 729
Table 22-13. FMRATE Register ..... 730
Table 22-14. SRCIF Register ..... 731
Table 22-15. SRCSIZE0 Register ..... 731
Table 22-16. SRCSIZE1 Register ..... 731
Table 22-17. SRCSIZE2 Register ..... 732
Table 22-18. SDSIZE Register ..... 732
Table 22-19. IEGAIN Register ..... 733
Table 22-20. IECTRL Register ..... 733
Table 22-21. YCPT0 Register ..... 733
Table 22-22. YCPT1 Register ..... 733
Table 22-23. YCPT2 Register ..... 734
Table 22-24. YCSLP0 Register ..... 735
Table 22-25. YCSLP1 Register ..... 735
Table 22-26. YCSLP2 Register ..... 735
Table 22-27. YCSLP3 Register ..... 736
Table 22-28. YCSLP1 Register ..... 736
Table 22-29. YCSLP5 Register ..... 736
Table 22-30. YLEVEL Register ..... 737
Table 22-31. MXCOE0 Register ..... 737
Table 22-32. MXCOE1 Register ..... 737
Table 22-33. SPTH Register ..... 738
Table 22-34. NLRSDIF Register ..... 739
Table 22-35. DDMA0 Register ..... 739
Table 22-36. UDMA0 Register ..... 739
Table 22-37. MEMSRC0 Register ..... 741
Table 22-38. MEMSRC1 Register ..... 741
Table 22-39. MEMSRCPIT Register ..... 742
Table 22-40. MEMSRC2 Register ..... 742
Table 22-41. DMA1 Register ..... 742

Table 22-42. MDEST0 Register ..... 742
Table 22-43. MDEST1 Register ..... 743
Table 22-44. MDEST2 Register ..... 743
Table 22-45. MDEST3 Register ..... 743
Table 22-46. MDEST4 Register ..... 743
Table 22-47. MDEST5 Register ..... 744
Table 22-48. OSDFONT Register ..... 744
Table 22-49. OSDDISP Register ..... 744
Table 22-50. OSDREAD Register ..... 746
Table 22-51. OSDEN Register ..... 746
Table 22-52. OSDPAT0 Register ..... 746
Table 22-53. OSDPAT1 Register ..... 747
Table 22-54. OSDPAT2 Register ..... 747
Table 22-55. OSDPAT3 Register ..... 747
Table 22-56. OSDPAT4 Register ..... 748
Table 22-57. OSDPAT5 Register ..... 748
Table 22-58. OSDPAT6 Register ..... 748
Table 22-59. OSDCOR0 Register ..... 749
Table 22-60. OSDWSZ0 Register ..... 750
Table 22-61. OSDSSZ0 Register ..... 750
Table 22-62. OSDFSZ0 Register ..... 750
Table 22-63. OSDCOR1 Register ..... 752
Table 22-64. OSDWSZ1 Register ..... 753
Table 22-65. OSDSSZ1 Register ..... 753
Table 22-66. OSDFSZ1 Register ..... 753
Table 22-67. OSDCOR2 Register ..... 754
Table 22-68. OSDWSZ2 Register ..... 755
Table 22-69. OSDSSZ2 Register ..... 755
Table 22-70. OSDFSZ2 Register ..... 755
Table 22-71. OSDCOR3 Register ..... 756
Table 22-72. OSDWSZ3 Register ..... 757
Table 22-73. OSDSSZ3 Register ..... 757
Table 22-74. OSDFSZ3 Register ..... 757
Table 22-75. INTSTS Register ..... 758

Table 22-76. INTMASK Register ..... 758
Table 22-77. SCARSTS Register ..... 759
Table 22-78. ITU-R BT. 656 EAV and SAV Sequence ..... 760
Table 23-1. Summary of UART Controller Registers ..... 772
Table 23-2. Receiver Buffer Register ..... 773
Table 23-3. Transmitter Holding Register ..... 773
Table 23-4. Interrupt Enable Register ..... 774
Table 23-5. Interrupt Identification Register ..... 774
Table 23-6. Interrupt Control Table ..... 775
Table 23-7. FIFO Control Register ..... 777
Table 23-8. FIFO Trigger Level of the Receiver ..... 778
Table 23-9. FIFO Trigger Level of the Transmitter ..... 778
Table 23-10. Line Control Register ..... 779
Table 23-11. Parity Setting Table ..... 780
Table 23-12. Word Length and Stop Bit Setting Table ..... 780
Table 23-13. Modem Control Register ..... 781
Table 23-14. Line Status Register ..... 782
Table 23-15. Testing Register ..... 784
Table 23-16. Modem Status Register ..... 784
Table 23-17. Scratch Pad Register ..... 785
Table 23-18. Baud Rate Divisor Latch LSB ..... 786
Table 23-19. Baud Rate Divisor Latch MSB ..... 786
Table 23-20. Prescaler Register ..... 786
Table 24-1. SSP Control Register Specification ..... 789
Table 24-2. SSP Control Register 0 ..... 790
Table 24-3. SSP Control Register 1 ..... 792
Table 24-4. SSP Control Register 2 ..... 793
Table 24-5. SSP Status Register ..... 794
Table 24-6. Interrupt Control Register ..... 795
Table 24-7. Interrupt Status Register ..... 796
Table 25-1. Summary of SD Host Controller Registers ..... 800
Table 25-2. Command Register (Offset: 0x00) ..... 801
Table 25-3. Argument Register (Offset: 0x04) ..... 802
Table 25-4. Response 0-3 Register (Offset: 0x08 ~ 0x14) ..... 802

Table 25-5. Short Responses and Long Responses (Offset: 0x08 ~ 0x14) ..... 802
Table 25-6. Responded Command Register (Offset: 0x18) ..... 803
Table 25-7. Bit Assignment of Data Control Register (Offset: 0x1c) ..... 803
Table 25-8. Data Block Size ..... 804
Table 25-9. Data Timer Register (Offset: 0x20) ..... 805
Table 25-10. Data Length Register (Offset: 0x24) ..... 805
Table 25-11. Status Register (Offset: 0x28) ..... 806
Table 25-12. Clear Register (Offset: 0x2c) ..... 807
Table 25-13 Interrupt Mask Register (Offset: 0x30) ..... 808
Table 25-14. Power Control Register (Offset: 0x34) ..... 809
Table 25-15. Clock Control Register (Offset: 0x38) ..... 809
Table 25-16. Bus Width Register (Offset: 0x3c) ..... 810
Table 25-17. Data Window Register (Offset: 0x40) ..... 811
Table 25-18. MMC Interrupt Response Time (Offset: 0x44) ..... 811
Table 25-19. General-purpose Output Register (Offset: 0x48) ..... 811
Table 25-20. SDIO Control Register (Offset: 0x6c) ..... 811
Table 25-21. SDIO Control Register (Offset: 0x70) ..... 812
Table 25-22. SDIO Status Register (Offset: 0x74) ..... 812
Table 25-23. Virtual Memory Register (Offset: 0X10000) ..... 813
Table 26-1. Summary of FTIIC010 I2C Controller Register ..... 818
Table 26-2. I2C Control Register ..... 819
Table 26-3. I2C Status Register ..... 821
Table 26-4. I2C Clock Division Register ..... 822
Table 26-5. I2C Data Register ..... 823
Table 26-6. I2C Slave Address Register ..... 823
Table 26-7. I2C Set/Hold Time and Glitch Suppression Setting Register ..... 824
Table 26-8. I2C Bus Monitor Register ..... 825
Table 26-9. I2C Revision Register ..... 825
Table 26-10. I2C Feature Register ..... 825
Table 26-11. SM Control Register ..... 826
Table 26-12. SM Maximum Timeout Register ..... 826
Table 26-13. SM Minimum Timeout Register ..... 827
Table 26-14. SM Master Extend Time Register ..... 827
Table 26-15. SM Slave Extend Time Register ..... 827

Table 27-1. FTGPIO010 Register Summary ..... 832
Table 27-2. GPIO Data-Out Register ..... 834
Table 27-3. GPIO Data In Register ..... 834
Table 27-4. Pin Direction Register ..... 835
Table 27-5. Pin Bypass Mode Register ..... 835
Table 27-6. GPIO Data Bit Set Register ..... 835
Table 27-7. GPIO Data Bit Clear Register ..... 836
Table 27-8. Pin Pull Enable Register ..... 836
Table 27-9. Pin Pull Type Register ..... 836
Table 27-10. Pull Truth Table ..... 837
Table 27-11. Interrupt Enable Register ..... 837
Table 27-12. Interrupt Raw State Register ..... 837
Table 27-13. Interrupt Masked State Register ..... 838
Table 27-14. Interrupt Mask Register ..... 838
Table 27-15. Interrupt Clear Register ..... 838
Table 27-16. Interrupt Trigger Method Register ..... 839
Table 27-17. Interrupt Both Edge Trigger Register ..... 839
Table 27-18. Interrupt Rising or Falling Edge Trigger Register ..... 839
Table 27-19. Bounce Enable Register ..... 840
Table 27-20. Bounce Clock Pre-scale Register ..... 840
Table 28-1. Interrupt Routing Table (For IRQ and FIQ) ..... 843
Table 28-2. Summary of Interrupt Controller Registers ..... 845
Table 28-3. IRQ Source Register (+0x00) ..... 847
Table 28-4. IRQ Enable Register (+0x04) ..... 847
Table 28-5. IRQ Interrupt Clear Register (+0x08) ..... 847
Table 28-6. IRQ Trigger Mode Register (+0x0C) ..... 848
Table 28-7. IRQ Trigger Level Register (+0x10) ..... 848
Table 28-8. IRQ Status Register (+0x14) ..... 848
Table 28-9. FIQ Source Register (+0x20) ..... 849
Table 28-10. FIQ Enable Register (+0x24) ..... 849
Table 28-11. FIQ Interrupt Clear Register (+0x28) ..... 849
Table 28-12. FIQ Trigger Mode Register (+0x2C) ..... 850
Table 28-13. FIQ Trigger Level Register (+0x30) ..... 850
Table 28-14. FIQ Status Register (+0x34) ..... 850

Table 28-15. Revision Register (+0x50) ..... 851
Table 28-16. Feature Register for Input Number (+0x54) ..... 851
Table 28-17. Feature Register for IRQ De-bounce Location (+0x58) ..... 851
Table 28-18. Feature Register for FIQ De-bounce Location (+0x5C) ..... 851
Table 28-19. Extended IRQ Source Register (+0x60) ..... 852
Table 28-20. Extended IRQ Enable Register (+0x64) ..... 852
Table 28-21. Extended IRQ Clear Register (+0x68) ..... 852
Table 28-22. Extended IRQ Trigger Mode Register (+0x6C) ..... 853
Table 28-23. Extended IRQ Trigger Level Register (+0x70) ..... 853
Table 28-24. Extended IRQ Status Register (+0x74) ..... 853
Table 28-25. Extended Feature Register for IRQ De-bounce Location (+0x78) ..... 854
Table 28-26. Extended FIQ Source Register (+0x80) ..... 854
Table 28-27. Extended FIQ Enable Register (+0x84) ..... 854
Table 28-28. Extend FIQ Interrupt Clear Register (+0x88) ..... 855
Table 28-29. Extended FIQ Trigger Mode Register (+0x8C) ..... 855
Table 28-30. Extended FIQ Trigger Level Register (+0x90) ..... 855
Table 28-31. Extended FIQ Status Register (+0x94) ..... 856
Table 28-32. Extended Feature Register for FIQ De-bounce Location (+0x78) ..... 856
Table 29-1. Summary of Timer Registers ..... 859
Table 29-2. Tm1Counter, Tm2Counter, Tm3Counter Registers ..... 860
Table 29-3. Tm1Load, Tm2Load, Tm3Load Registers ..... 860
Table 29-4. Tm1Match1, Tm2Match1, Tm3Match1 Registers ..... 861
Table 29-5. Tm1Match2, Tm2Match2, Tm3Match2 Registers ..... 861
Table 29-6. Timer Control Register ..... 862
Table 29-7. Timer Interrupt State Register ..... 863
Table 29-8. Timer Interrupt Mask Register ..... 865
Table 29-9. TmRevision Register ..... 866
Table 30-1. Summary of Timer Registers ..... 870
Table 31-1. Summary of WatchDog Timer Register ..... 883
Table 31-2. WdCounter Register ..... 884
Table 31-3. WdLoad Register ..... 884
Table 31-4. WdRestart Register ..... 884
Table 31-5. WdCR Register ..... 885
Table 31-6. WdStatus Register ..... 885

Table 31-7. WdClear Register ..... 885
Table 31-8. WdIntrlen Register ..... 886
Table 31-9. WdRevision Register ..... 886
Table 32-1. Crossing Clock Domain Signals ..... 891
Table 32-2. RTC Register Summary ..... 892
Table 32-3. RtcSecond Register ..... 893
Table 32-4. RtcMinute Register ..... 893
Table 32-5. RtcHour Register ..... 894
Table 32-6. RtcDays Register ..... 894
Table 32-7. AlarmSecond Register ..... 894
Table 32-8. AlarmMinute Register ..... 895
Table 32-9. AlarmHour Register ..... 895
Table 32-10. Control Register of RTC ..... 896
Table 32-11. WRtcSecond Register ..... 897
Table 32-12. WRtcMinute Register ..... 897
Table 32-13. WRtcHour Register ..... 898
Table 32-14. WRtcDay Register ..... 898
Table 32-15. IntrState Register ..... 898
Table 32-16. RtcDivide Register ..... 899
Table 32-17. RtcRevision Register ..... 899
Table 32-18. RtcCurrent Register ..... 900
Table 33-1. Absolute Maximum Ratings ..... 902
Table 33-2. Recommended Operating Conditions ..... 902
Table 33-3. I/O Pad Capacitance ..... 902
Table 33-4. DC Characteristics of 3.3 V I/O Cells ..... 903

# LIST OF FIGURES


Figure 1-1. Functional Block Diagram ..... 8
Figure 2-1. TFBGA-256 Pin Assignments (Top View) ..... 36
Figure 3-1. Physical Memory Map of GM8126 ..... 38
Figure 4-1. PMU Block Diagram for PLL32K, PLL1, and PLL2 ..... 41
Figure 6-1. Block Diagram of AHB Controller ..... 115
Figure 7-1. Functional Block Diagram of APB Bridge ..... 123
Figure 7-2. Hardware Handshake Protocol ..... 132
Figure 8-1. Functional Block Diagram ..... 134
Figure 8-2. Connection Scheme of Asynchronous Devices ..... 136
Figure 8-3. Timing Diagram of SMC Write to Asynchronous Devices ..... 140
Figure 8-4. Timing Diagram of SMC Read from Asynchronous Devices ..... 142
Figure 8-5. Timing Diagram of SMC Write to VLIO Devices ..... 144
Figure 8-6. Timing Diagram of SMC Read from VLIO Devices ..... 144
Figure 9-1. Functional Block Diagram of DDR2 Memory Controller ..... 149
Figure 9-2. Group-level Re-arbitration ..... 153
Figure 9-3. Basic DDR2 RC Timing Parameter ..... 155
Figure 10-1. Functional Block Diagram of DMA Controller Without Simple Bridge Configuration ..... 199
Figure 10-2. Arbitration Scheme ..... 201
Figure 10-3. Linked List Structure of Chain Transfer Operation ..... 202
Figure 10-4. Linked List Descriptor ..... 202
Figure 10-5. Example of Interrupt Conflict During Chain Transfer Operation ..... 204
Figure 10-6. LLP_CNT Operation ..... 204
Figure 10-7. Example of Hardware Handshake Mode Transfer ..... 206
Figure 10-8. DMA Hardware Handshake Protocol ..... 207
Figure 10-9. ERROR Response Happens When in Hardware Handshake Mode Transfer ..... 207
Figure 10-10. Abort Operation When Channel is Active ..... 207
Figure 10-11. Abort Operation When Channel is Inactive ..... 208
Figure 10-12. Example of DMA Normal Mode Transfer ..... 209
Figure 11-1. Block Diagram ..... 242

Figure 12-1. Block Diagram ..... 259
Figure 13-1. Block Diagram of FTMCP210 ..... 307
Figure 14-1. FTMCP100 Block Diagram ..... 314
Figure 15-1. Block Diagram ..... 322
Figure 15-2. Little-endian Byte and Little-endian Pixel (LBLP) ..... 324
Figure 15-3. Big-endian Byte and Big-endian Pixel (BBBP) ..... 325
Figure 15-4. Little-endian Byte and Big-endian Pixel (LBBP) ..... 326
Figure 15-5. Raw Data Translation for RGB444 and RGB555 Memory Formats ..... 328
Figure 15-6. Palette Remapping Operation ..... 329
Figure 15-7. Font Structure (12 x 16 Dots Per Font) ..... 330
Figure 15-8. OSD Basic Operation Description ..... 331
Figure 15-9. Block Diagram of Scaler ..... 332
Figure 15-10. 1/2x1/2 Down-scaling ..... 332
Figure 15-11. Nearly Bilinear Interpolation ..... 333
Figure 15-12. Threshold Nearly Bilinear Interpolation ..... 334
Figure 15-13. Threshold Nearly Bilinear Interpolation ..... 334
Figure 15-14. Vertical Timing of BT656/BT1120 ..... 349
Figure 15-15. Horizontal Timing of BT656/BT1120 ..... 350
Figure 16-1. Transmit Ring Descriptor Structure ..... 392
Figure 16-2. Receive Ring Descriptor Structure ..... 395
Figure 16-3. Example of Incoming Packet Placement ..... 398
Figure 17-1. Block Diagram of FTTVE100 Controller ..... 415
Figure 18-1. Block Diagram ..... 423
Figure 19-1. Block Diagram ..... 439
Figure 19-2. Timing of sync_reg_update ..... 446
Figure 19-3. Relationship between Source Image and Target Image ..... 448
Figure 19-4. VPSRC_X and VP_SRC_Y ..... 457
Figure 19-5. Source Active Window ..... 458
Figure 19-6. Window Mask ..... 461
Figure 19-7. Vertical and Horizontal Flip Address Control Example ..... 484
Figure 19-8. Start Address of Destination Frame Buffer ..... 488
Figure 19-9. OSD Font RAM and Display RAM ..... 492
Figure 19-10. Example of OSD ..... 499
Figure 19-11. Vertical interval of ITU-R BT. 656 ..... 523

Figure 19-12. Timing of ITU-R BT. 656 Like ..... 524
Figure 19-13. Vertical Timing of 8-bit H/V Reference Control ..... 524
Figure 19-14. Horizontal Timing of 8-bit H/V Reference Control ..... 525
Figure 19-15. Vertical Timing of 16-bit H/V Sync. Control ..... 525
Figure 19-16. Horizontal Timing of 16-bit H/V Sync. Control ..... 525
Figure 19-17. Independent Two-path Size-down ..... 526
Figure 20-1. Block Diagram ..... 531
Figure 20-2. Color Correction Matrix ..... 557
Figure 20-3. AE STA Window Gap ..... 569
Figure 20-4. AE STA Threshold ..... 570
Figure 20-5. Frame/Line Sync. Related Status ..... 597
Figure 20-6. SVREF_DEF = 1 and SHREF_DEF = 1 ..... 622
Figure 20-7. SVREF_DEF = 0 and SHREF_DEF = 0 ..... 622
Figure 20-8. SVREF_DEF = 1 and SHREF_DEF = 0 ..... 623
Figure 20-9. RGB Gain ..... 624
Figure 20-10. Resizing Down ..... 625
Figure 20-11. Aspect Ratio on Sizing Down ..... 626
Figure 20-12. Register and Table Access ..... 627
Figure 21-1. Block Diagram ..... 631
Figure 21-2. Timing Diagram of Address State ..... 648
Figure 21-3. Timing Diagram of Command State ..... 649
Figure 21-4. Timing Diagram of Write Data State ..... 649
Figure 21-5. Timing Diagram of Read Data State ..... 650
Figure 21-6. Timing Diagram of Busy State ..... 650
Figure 21-7. Timing Diagram of Buffer State ..... 651
Figure 21-8. Timing Diagram of Address State ..... 652
Figure 21-9. Timing Diagram of Pre-amble Write State ..... 652
Figure 21-10. Timing Diagram of Pre-amble Read State ..... 653
Figure 21-11. Timing Diagram of Post-amble Write State ..... 653
Figure 21-12. Timing Diagram of Post-amble Read State ..... 654
Figure 21-13. Timing Diagram of Pre-amble Write State ..... 654
Figure 21-14. Timing Diagram of Pre-amble Read State ..... 655
Figure 21-15. Timing Diagram of Post-amble Write State ..... 655
Figure 21-16. Timing Diagram of Post-amble Read State ..... 656

Figure 21-17. Summary of AC Timing Usage ..... 657
Figure 21-18. Data SRAM Address Offset Mapped to BMC Region ..... 681
Figure 21-19. Mapping Between AHB Slave and BMC Region 0 ..... 682
Figure 21-20. User Mode Operation ..... 683
Figure 21-21. Sector Mode Data Format ..... 686
Figure 21-22 Page Mode Data Format ..... 687
Figure 21-23. 2-plane Write 4 Pages Cross Block ..... 696
Figure 21-24. Spare SRAM Organization of Command Queue Depth is 2 ..... 698
Figure 21-25. Spare SRAM Address Mapping ..... 699
Figure 21-26. Spare Data Position of 16 Sectors ..... 699
Figure 21-27. Spare Data Position of Sector Offset 1 ..... 699
Figure 21-28. Spare Data Position of 8 Sectors ..... 700
Figure 21-29. Spare Data Position of Sector Offset 6 ..... 700
Figure 21-30. Spare Data Position of 4 Sectors ..... 701
Figure 21-31. Spare Data Position of 2 Sectors ..... 701
Figure 21-32. Spare Data Position of 1 Sector ..... 701
Figure 21-33. Spare Data Position in Page Mode ..... 702
Figure 21-34. User Data, CRC, and ECC Usages ..... 702
Figure 22-1. Block Diagram ..... 717
Figure 22-2. Video Capture Path ..... 718
Figure 22-3. Loopback Path ..... 718
Figure 22-4. Timing of sync_reg_update ..... 723
Figure 22-5. Relationship between Source Image and Target Image ..... 724
Figure 22-6. Relationship between Border Width/Height and Crop Window ..... 726
Figure 22-7. Source Active Window ..... 732
Figure 22-8. Luma Curve ..... 734
Figure 22-9. Color Matrix ..... 738
Figure 22-10. OSD Font RAM and Display RAM ..... 745
Figure 22-11. OSD Example ..... 751
Figure 22-12. Vertical interval of ITU-R BT. 656 ..... 760
Figure 22-13. Timing of ITU-R BT. 656 Like ..... 761
Figure 22-14. Macro Block Order in One Image ..... 763
Figure 22-15. Pixel Output Order in One 16 x 16 Macro Block ..... 764
Figure 22-16. Pixel Output Order in One 8 x 8 Macro Block ..... 765

Figure 22-17. Pixel Output Order in One 4 x 4 Macro Block ..... 766
Figure 22-18. YCbCr 4:2:0 Macro Block Order in Two-split Memory Block ..... 767
Figure 23-1. Block Diagram of UART Controller ..... 771
Figure 23-2. Interconnection between MCR and MSR in Loopback Mode ..... 782
Figure 24-1. Functional Block Diagram ..... 789
Figure 25-1. Block Diagram ..... 799
Figure 26-1. Functional Block Diagram ..... 817
Figure 26-2. Relationship among TSR, SCL, and SDA ..... 824
Figure 27-1. Block Diagram ..... 831
Figure 28-1. Functional Block Diagram ..... 843
Figure 29-1. Block Diagram ..... 858
Figure 30-1. Pin Diagram ..... 869
Figure 30-2. Example of Timer Operation ..... 876
Figure 30-3. Double Buffering Function ..... 877
Figure 30-4. Example of PWM ..... 878
Figure 30-5. Waveform of Enabled Dead-zone Feature ..... 879
Figure 30-6. DMA Request and ACK ..... 879
Figure 31-1. Block Diagram of the WatchDog Timer ..... 882
Figure 32-1. Block Diagram ..... 889
Figure A-1. Package Outline of E-PAD TQFP-176 ..... 905
Figure A-2. Package Outline of TFBGA-256 ..... 906

# Chapter 1

# Introduction

This chapter contains the following sections:


- 1.1 Version of the Chip
- 1.2 Features
- 1.3 Block Diagram

The GM8126 video chip with the FA626TE CPU core provides a high-performance solution to accelerate the image or video applications, such as H.264, MPEG4, and JPEG, of the end products. GM8126 supports the JPEG, MPEG4, and H.264 encoders. It provides a cost-effective and easy development system for the integration and verification of the video applications at the early development stage. The GM8126 system platform includes a wide range of basic components, which include the H.264 encoder, video capture, display controller, DES/3DES controller, USB 2.0 OTG controller, USB 2.0 PHY, DMA controller, Ethernet MAC controller, Ethernet PHY, and AMBA bus framework, to reduce the overall system cost.

## 1.1 Version of the Chip

This is the preliminary version of the chip.

## 1.2 Features

**Architecture**

- AMBA AHB bus for high-speed devices
- AMBA APB bus for low-power devices

**Embedded Processor**

- Faraday FA626TE
- ARM-based 32-bit RISC
- Up to 533 MHz
- 16 KB instruction cache and 8 KB data cache

**Memory Interface**

- 8-bit/16-bit DDR2 SDRAM interface
- Supports NAND/SPI flash boot

**High-speed I/O Interface**

- USB 2.0 OTG
- MII/RMII

**Capture Interface**

- ITU BT-656 input X2 (Up to 108 MHz)
- Bayer raw data input X1 (Up to 108 MHz)

**Display Interface**

- NTSC/PAL composite output
- ITU BT.1120 digital output (74.25 MHz)

**Clock / PLL**

- 32768 Hz and 30 MHz (Optional) oscillators
- Programmable frequency of core PLL for main clock and audio main clocks
- Programmable clock output X2 (25 MHz/27 MHz/54 MHz/108 MHz)
- Clock gating control for individual components

**Power Management**

- Frequency change control
- Normal operation
- Idle-mode operation
- Standby-mode operation
- Sleep-mode operation

**AHB Bus Components**

- DDR2 SDRAM controller
- 32-bit static memory controller
- DMA controller
- USB 2.0 OTG controller
- Video capture (NTSC/PAL)
- 3D-deinterlace and de-noise filter
- LCD controller (Max. 1024 x 768)
- H.264 encoder
- MEPEG4 codec
- JPEG codec
- AES/DES/TDES cipher controller
- 10/100 Ethernet MAC controller

- SDIO/SDHC

**APB Bus Components**

- Timer
- Watchdog timer
- Real time clock
- Interrupt controller
- GPIO x 4
- $I^2C$
- $I^2S$
- Full-function UART x 1 and Console UART x 1 (Or Console UART x 5)

**H.264 Encoder**

- Supports baseline profile encoding of up to D1 at 120 fps (Max.)

**JPEG Codec**

- Up to D1 at 60 fps

**3D-deinterlace and De-noise filter**

- Built-in de-noise filter to eliminate high-frequency noise of field difference
- Supports macro-block-based picture with maximum resolution of up to 720 x 480i
- Supports four-field motion detection (Forward forward/forward/current/next)
- Supports only macro-block planar Y format for YcbCr 444/422/420/400 source
- Supports macro-block semi-planar YcbCr format for YcbCr 422/420 source
- Supports raster packet YcbCr422 format for packet YcbCr 422 source

**Video Capture Interface**

- Supports maximum capture resolution of up to 1920x1080
- Supports two BT.656 input ports up to 80 MHz
- Supports BT.1120 input ports up to 80 MHz
- YCbCr 4:2:2 8-bit/16-bit with H/V reference control signal interface
- One video capture input with two data output paths
- Edge-based line-average de-interlacer
- Supports size-down function at two output paths

- Path 0: Supports integer sizing-down ratio
- Path 1: Supports fraction sizing-down ratio

- Supports up to 8 mask windows with transparency degree control
- Supports font-based OSD at two output ports
- Individual frame-skip function at two output paths
- Individual even/odd field-sorting function at two output paths
- Source and output target image crop
- Supports the YCbCr 4:2:2 format at two output ports
- DMA rater order output sequence to support output frame flip function
- Provides clock output to video decoder to save external oscillator (27 MHz, 54 MHz, 108 MHz)

**Image Sensor Processor**

- Supports resolution of up to 2592x1944
- Frequency of pixel clock up to 108MHz
- Active pixel processing ability: 80 Mega pixels/sec
- Supports 8-bit/10-bit/12-bit RGB-Bayer raw data input
- Supports optical black calibration
- Supports dark frame subtraction
- Supports lens shading compensation
- Supports auto defect pixel correction
- Supports edge-adaptive color interpolation
- Supports programmable color correction matrix
- Supports programmable RGB to YCbCr color space conversion matrix
- Supports individual RGB gamma table for gamma correction
- Supports luminance compensation curve
- Supports non-linear color saturation and hue control
- Supports contrast enhancement to lighten shadows and darken highlights
- Supports false color and chroma suppression
- Supports programmable sharpness gain & threshold control
- Supports linear seamless image resize down
- Supports image mirror, flip, and cropping
- Supports AE/AWB/AF/Motion detection statistics
- Supports Y/R/G/B histogram

**Display Controller**

- Supports 1024 x 768 resolutions
- Integrates 27-MHz DAC for composite output (NTSC/PAL)
- Supports font-based OSD for display outputs

**Embedded Analog Device**

- USB 2.0 OTG PHY
- DDR2 PHY
- 2.5-V output for DC-DC linear regulator
- Ethernet 10/100 Fast Ethernet transceiver
- 10-bit resolution DAC
- Stereo audio codec
- Digital Microphone input
- PLL x 3

**Operating Frequency**

- AMBA AHB bus: Up to 266 MHz
- AMBA APB bus: Up to 133 MHz
- DDR2: Up to 533 MHz

**Operating Voltage**

- 1.0 V ±5% for core power
- 3.3 V ±5% for inputs and outputs
- 1.8 V ±5% for DDR2 DRAM I/Os

**Operating Temperature Range**

- From 0 °C to 75 °C

**Process**

- UMC 90 nm logic process

**Package**

- 256-pin TFBGA
- 176-pin E-PAD TQFP

## 1.3 Block Diagram

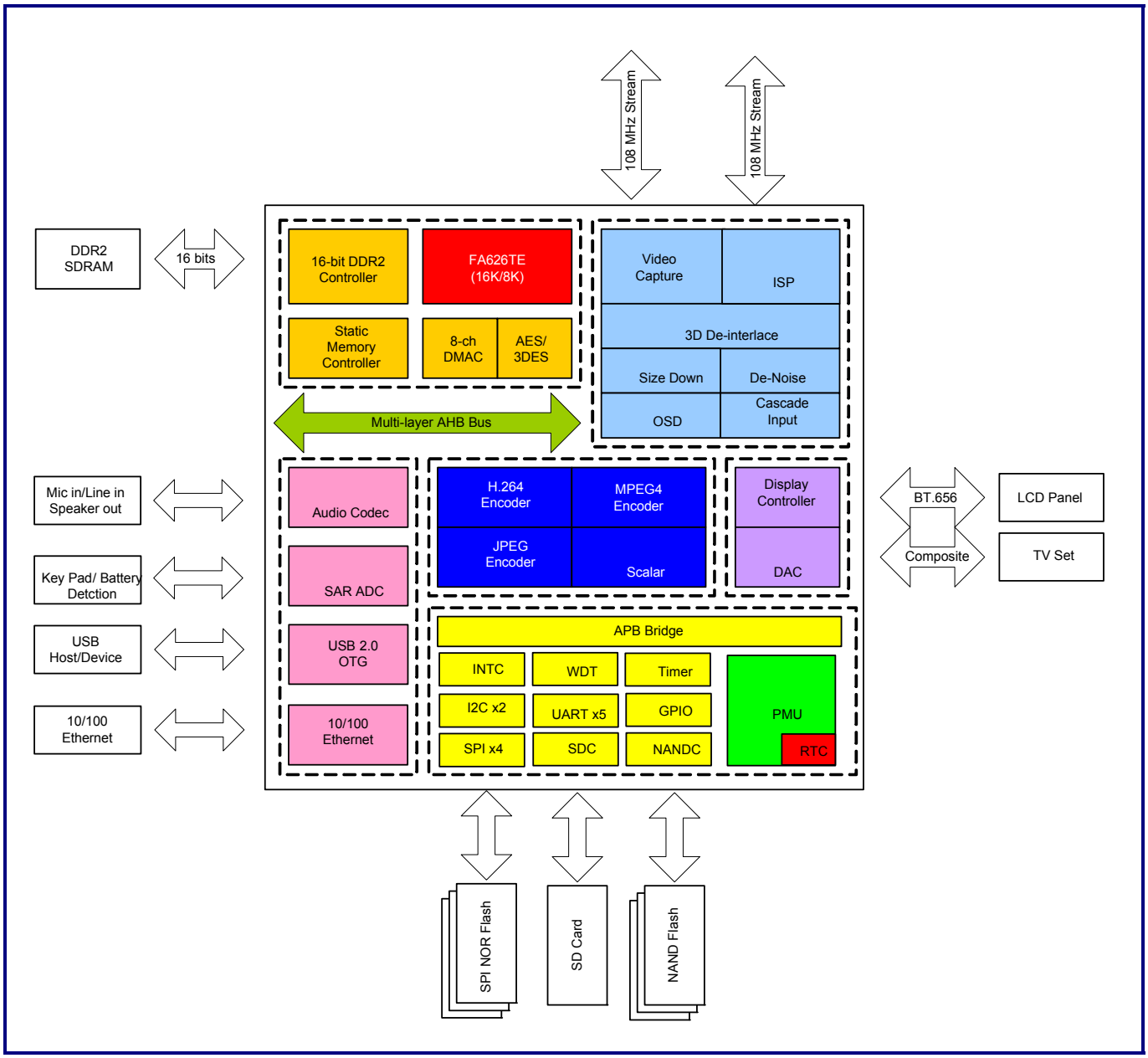



**Figure 1-1. Functional Block Diagram**

### 1.3.1 FA626

FA626 is an ultra high-speed general-purpose 32-bit embedded RISC processor. It includes a CPU core, a 16-kB instruction cache, an 8-kB data cache, an instruction scratchpad, a data scratchpad, a write buffer, a memory management unit, and a JTAG ICE interface.

The CPU core is a Harvard architecture design with eight pipeline stages, which include the Fetch, Instruction, Decode, Register Access, Shift, Execution, Memory, and Write stages. To reduce the branch penalties, FA626 contains a Branch Target Buffer (BTB) and a Return Stack (RS) to improve its overall performance. FA626 MMU implements an enhanced ARM architecture and a V4 MMU to provide the address translations and the permission check mechanism for accessing memories. FA626 operates at a speed of up to 533 MHz with low operating power.

### 1.3.2 AHB Controller (AHBC)

An AHB controller is used to support a mechanism for accessing the Advanced High-performance Bus (AHB). This mechanism contains an arbiter, a decoder, and a multiplexer, that are highly configurable. It also provides a flexibly programmable model for easy SoC integration. The features of AHBC include:

- Compliant with AMBA 2.0 standard
- Multi-level arbitration
- Round-robin arbitration

### 1.3.3 DDR2 SDRAM Controller (DDRC)

The DDR2 SDRAM controller supports rich types of DDR2 RAMs and uses a DDR2 burst length of four to accelerate the read/write speeds. The features of DDRC include:

- Supports rich types of DDR2 RAMs
- Uses 16-bit data bus width
- Supports programmable refresh types (Staggered or non-staggered)
- Supports DDR power-down function (DDR RAM in the self-refresh mode)
- Pre-fetches sequential read data for AHB burst read command
- 16-word data FIFO for each channel to enhance the performance of a read/write command
- Supports two types of MA tables

### 1.3.4 Static Memory Controller (SMC)

SMC supports ROM. The features of SMC include:

- 32-bit data width
- Supports up to three banks
- Supports memory size of up to 256 MB for each bank
- Supports flexible host interface with ready signal
- Shadows first bank with other banks

### 1.3.5 NAND Flash Controller (NANDC)

The NAND Flash controller provides an interface to access the NAND Flash. It provides a hardware-based NAND host controller that transfers data in the high-speed mode. The ECC information is automatically generated in the programming operation and is checked in the read operation. The features of NANDC include:

- Supports interleaving operation
- Supports up to 24 bits of ECC correction capability
- Supports programmable NAND Flash timing parameters
- Supports programmable NAND Flash command flow
- Supports DMA handshake mode
- Supports up to 2 CEs (Flash chip enable)
- Supports data bus width: 8 bits
- Supports page sizes of 512, 2k, 4k and 8k bytes for NAND Flash

### 1.3.6 Direct Memory Access Controller (DMAC)

DMAC is used to enhance the system performance. The system efficiency is improved with the high-speed data transfers among the system and devices by reducing the processor-interrupt generation. DMAC provides up to seven channels for the memory-to-memory, memory-to-peripheral, and peripheral-to-memory transfers with the shared buffers. The features of DMAC include:

- Provides memory-to-memory, memory-to-peripheral, and peripheral-to-memory transfers
- Group round-robin arbitration scheme with four priority levels
- Supports chain transfer
- Supports transfers with 8-bit, 16-bit, or 32-bit data width

### 1.3.7 USB 2.0 OTG Controller

The USB 2.0 On-The-Go (OTG) controller plays dual roles as a host controller or as a peripheral controller. When acting as a host, it contains a USB host controller that supports the transactions in all speed modes. Without the software intervention, the host controller can deal with the transaction-based data structure to offload CPU and automatically transmit and receive data on the USB bus. When acting as a peripheral controller, each endpoint, except for endpoint 0, accepts the programmable HS or FS transfers to suit all applications. In addition, being compliant with the OTG standards means that both the Session Request Protocol (SRP) and Host Negotiation Protocol (HNP) are available. The transceiver interface is UTMI+ Level 2, which supports the HS, FS, and LS transfers without a hub.

The features of the USB 2.0 OTG controller include:

- Compliant with USB Specification, Revision 2.0
- Compliant with On-The-Go supplement of USB 2.0 Specification, Revision 1.0
- Supports UTMI+ Level 2 compliant transceiver
- Compatible with EHCI 1.0
- Supports OTG SRP and HNP protocols
- Supports point-to-point communications with one HS, FS, or LS device
- Hardware configurable endpoints to act as HS and FS devices
- Supports isochronous, interrupt, control, and bulk transfers for both host and device
- Compatible with EHCI data structures
- Embedded DMA for accessing FIFO
- SRAM interface for FIFO
- Supports suspend, remote wake-up, and resume functions
- Completely synchronous design
- USB OTG software subsystem

### 1.3.8 H.264 Encoder

FTMCP210 is a hardware video encoder for the baseline profile of the MPEG4 AVC/JVT/H.264 video coding standard. This encoder is designed to transform a sequence of YcbCr 4:2:0 pictures into a compressed video bitstream. The supported resolution is ranging from 128 x 80 to 1920 x 1088 in steps of 16 units. For better quality, this encoder supports various block sizes for inter-prediction. The motion vector is quarter-pel accurate for each block type. The search ranges is ±24 in the horizontal direction and ±16 in the vertical direction. The clock rate of encoding a SDTV sequence in the real time is below 54 MHz. The features of FTMCP210 include the following MPEG4 AVC/JVT/H.264 (ISO/IEC 14496-10) video coding standards:

- Baseline profile level 3.1
- I and P frame encodings
- CBR and VBR (Rate control by firmware)
- Programmable in-loop filter parameter
- Programmable Chroma QP index offset parameter

### 1.3.9 MPEG4/JPEG Engine

The MPEG4/JPEG engine is an AHB-based codec. This codec includes the hardware engines to accelerate the computation intensive tasks, such as the motion estimation, DCT/IDCT, quantization/inverse quantization, and motion compensation. FTMCP100 is controlled by CPU through the AHB slave interface. By initializing the control registers of this codec, FTMCP100 will automatically perform the motion estimation calculation task for 16 x 16 or 8 x 8 block. FTMCP100 will also automatically perform the DCT/quantization, IDCT/inverse quantization, AC/DC prediction, zigzag scan, and VLC/VLD calculation task for a macro-block. Therefore, CPU can be released from the timing critical tasks in the video encoding. The standard-cell-based approach allows users to quickly integrate FTMCP100 into the SOC designs.

The features of the MPEG4/JPEG engine include:

- Compliant with MPEG4 (ISO/IEC 14496-2) simple profile L0 ~ L3 standards with resolutions of subQCIF, QCIF, CIF, VGA, and D1 at 60 fps with steps of 16 units
- Compliant with JPEG (ISO/IEC 10918-1) baseline standard
- Includes hardware engines for motion estimation/motion compensation, DCT/IDCT, quantization/inverse quantization, AC/DC prediction, and coding/decoding with variable length

- Local memory controller for controlling local memory shared by CPU, FTMCP100, and DMA master
- DMA controller to control data transfers between system memory and local memory
- Automatic power-down mechanism to reduce power consumption
- Motion estimation search range: -16 ~ +15.5 (Optionally -32 ~ +31) with half-pel accuracy
- Supports 4 MV and unrestricted MV
- Supports constant and variable bit rate controls of error resilient tools: Encoder supporting resynchronization marker and header extension code and decoder supporting resynchronization marker, header extension code, data partition, and RVLC
- Supports short video header (H.263 baseline)
- Supports H.263/MPEG/JPEG quantization methods

**JPEG features**:

- Supports four user-defined Huffman tables (2 AC and 2 DC)
- Supports four programmable quantization tables
- Supports interleave and non-interleave scans
- Supports YUV 4:4:4, 4:2:2, and 4:2:0 formats
- Supports image size of up to 64k × 64k
- Supports full-duplex operation (e. g. video phone and video conference) by S/W to switch encoding and decoding tasks on the same H/W

### 1.3.10 3D De-interlace De-noise Filter

The de-interlacing process is an important stage at which the ordinary TV interlaced sequences are converted into the progressive sequences for displaying on the progressive devices (Such as computer, LCD, plasma display, and projection TV).

The de-noise process is critical when enhancing the scene quality and reducing the encoded bit rate in network. In a real-time IP camera system, the de-noise function is very useful in eliminating the Gauss-distribution noise so that only small effort is required by the encoder for encoding the source scene. On the other hand, the encoded files will be smaller enough to reduce the bit rate transferred in network. The 3D de-interlace and de-noise module adopts the pixel-based measurement to perform the intra and inter field interpolations, depending on the motion state which the pixel belongs to particular motion or stationary. Its goal is to generate a high-quality progressive frame. The features include:

- Built-in de-noise filter to eliminate high-frequency noise of the field difference
- Supports macro-block-based picture with maximum resolution of up to 720 x 480i
- Supports four-field motion detection (Forward forward/forward/current/next)
- Supports only macro-block planar Y format for YcbCr 444/422/420/400 source
- Supports macro-block semi-planar YcbCr format for YcbCr 422/420 source
- Supports raster packet YcbCr422 format for YcbCr 422 source

### 1.3.11 Video Capture

The video capture is in charge of capturing video data from the AMBA AHB bus to the memory through the ITU-R BT. 656 interface. It provides the 3D de-interlace function to reduce the video artifact for the interlace video. The noise reduction function can remove undesired noise and preserve fine details and edges. With the size-down feature, users can reduce the size of the image to the individual resolution needed for the preview and record path. The color OSD function in the record path can help users paste any character at the captured video. The features of the video capture include:

- Maximum capture resolution: Up to 1920 x 1080
- Supports two BT.656 input ports of up to 108 MHz
- Supports BT.1120 input ports of up to 108 MHz
- YCbCr 4:2:2 8-bit/16-bit with H/V reference control signal interface
- One video capture input with two data output paths
- Edge-based line-average de-interlacer
- Supports size-down function at two output paths
  - Path 0: Supports integer sizing down ratio
  - Path 1: Supports fraction sizing down ratio
- Supports up to 8 mask windows with transparency degree control
- Supports font-based OSD at two output paths
- Individual frame-skip function at two output paths
- Individual even/odd field-sorting function at two output paths
- Source and output target image crop
- YCbCr 4:2:2 output format at two output paths
- DMA rater order output sequence and output frame flip and mirror function
- Provides clock output to video decoder to save external oscillator (27 MHz, 54 MHz, and 108 MHz)

### 1.3.12 Image Sensor Processor

FTISP210 is compatible with various CMOS image sensors. It provides a sensor compensation function, which is capable of compensating the black level and lens shading phenomenon. The proprietary color interpolation algorithm implemented in this IP will reconstruct the RGB components of every pixel from the Bayer raw data. The contrast enhancement, sharpness, false color suppression, and image enhancement control help users make mages more pleasing to the human eyes. Moreover, with consecutive size-down capability, users will resize the source image to the resolution needed. The image statistics on Auto-Focus (AF), Auto-Exposure (AE), Auto-White Balance (AWB), histogram, and motion detection can be further read for 3A or other image controller. The host processor interface is compliant with AMBA AHB 2.0, which serves as an IP core that can be integrated into the AMBA system. The raw data can also be read through AHB from other AHB slave for the purpose of beautification, resize, or other applications. The features of the image sensor processor includes:

- Supports resolution of up to 2592x1944
- Frequency of pixel clock up to 108MHz
- Active pixel processing ability: 80 Mega pixels/sec
- Supports 8-/10-/12-bit RGB-Bayer raw data input
- Supports optical black calibration
- Supports dark frame subtraction
- Supports lens shading compensation
- Supports auto defect pixel correction
- Supports edge-adaptive color interpolation
- Supports programmable color correction matrix
- Supports programmable RGB to YCbCr color space conversion matrix
- Supports Individual RGB gamma table for gamma correction
- Supports luminance compensation curve
- Supports non-linear color saturation and hue control
- Supports contrast enhancement to lighten shadows and darken highlights
- Supports false color and chroma suppression
- Supports programmable sharpness gain & threshold control
- Supports linear seamless image resize down
- Supports image mirror, flip, and cropping
- Supports AE/AWB/AF/Motion detection statistics

### 1.3.13 LCD Controller (LCDC)

The LCD controller acquires the video data from the frame buffer and output to provide all necessary control signals for various TFT LCD monitors or TV encoders. The features of the LCD controller include:

- Supports resolution of up to 1024 x 768
- Cursor
- Simple OSD

### 1.3.14 10/100 Mbps Ethernet Controller (MAC)

MAC is a high-quality Ethernet controller with the DMA function. It is composed of an AHB wrapper, a DMA engine, on-chip memories (TX FIFO and RX FIFO), MAC, and RMII interface.

MAC is an Ethernet controller that provides the AHB master capability and is fully compliant with the IEEE 802.3 Specification for 10/100 Mbps Ethernet. The FTMAC100_S Ethernet controller with the DMA function handles all data transfers between the system memory and the on-chip memories. With the DMA engine, this controller can reduce the CPU loading, maximize the performance, and minimize the FIFO size. FTMAC100_S contains the on-chip memories for buffering so that the external local buffer memory is not needed. The MII/RMII interfaces support two specific data rates, 10 Mbps and 100 Mbps. MAC provides the Wake-On-LAN function to support three wake-up events: Link status change, magic packet, and wake-up frame. This function allows systems that contain MAC to be woken up by the remote side. With the AHB wrapper, users can quickly integrate MAC into the SoC design. The features of MAC include:

- Provides DMA engine for transmitting and receiving packets
- Supports zero-copy data transfer
- Supports programmable AHB burst size
- Supports transmit and receive interrupt mitigations
- Supports Wake-On-LAN function and three wake-up events: Link status change, magic packet, and wake-up frame
- Supports four Wake-On-LAN signals (Active high, active low, positive pulse, and negative pulse)
- Contains independent TX/RX FIFOs (2 KB each)
- Supports half-duplex and full-duplex modes
- Supports flow control for full-duplex mode and backpressure for half-duplex mode
- Supports RMII interface

### 1.3.15 AES/DES/TDES Cipher Controller

The AES/DES/TDES cipher controller provides efficient hardware implementation of the DES/Triple-DES/AES algorithm for the high-performance encryption and decryption, which can be applied to various applications. Additionally, this controller provides the DMA function, which can reduce the data transfer overhead on the processor and improve the system performance. The features of the AES/DES/TDES cipher controller include:

- DES/Triple-DES encryption/decryption compliant with NIST Standard
- AES 128/192/256-bit encryption/decryption compliant with NIST Standard
- Block cipher mode supports the following modes:
  - DES/Triple-DES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
  - AES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
    - CTR mode
- Provides the DMA function

### 1.3.16 AHB-to-APB Bridge

An AMBA APB implementation typically contains a single APB bridge to convert an AHB transfer into a suitable format for the slave devices on the APB. The bridge latches all addresses, data, and control signals, as well as a second-level decoding to generate the slave-select signals for the APB peripherals. This APB Bridge is the only bus master on AMBA APB and is a slave on the higher level system bus. The APB Bridge provides a DMA function to enhance the performance of the data transfer. The features of the AHB-to-APB Bridge include:

- Supports four sets of independent DMA channels for the APB-to-AHB, AHB-to-APB, AHB-to-AHB, or APB-to-APB transfers
- Supports up to 14 sets of APB devices

### 1.3.17 Timer

The Faraday timer provides three independent sets of sub-timers. Each sub-timer can use the system clock (OSC) for increment or decrement counting. Two match registers are provided for each sub-timer. Whenever the value of the match register equals to any one of the sub-timers, the timer interrupt will be triggered immediately. The issuance of a timer interrupt can be decided by the register setting when the overflow occurs. The features of the timer include:

- Three independent 32-bit timer programming models
- Interrupts issued upon overflow and time-up
- Each sub-timer has two match registers
- Programmable decrementing/incrementing modes on the counter

### 1.3.18 Pulse Width Modulation (PWM)

PWM provides up to eight independent sets of timers. Each timer can use either the internal system clock (PCLK of APB) or external clock. Not only can these eight timers be used to generate the internal interrupts to CPU but also to trigger the DMA transfers. In addition, each timer supports the PWM (Pulse Width Modulation) function, which can generate the PWM signals for motor control or power level control.

### 1.3.19 Watchdog Timer (WDT)

The Watchdog timer (WDT) is used to prevent the system from infinite looping when the software becomes trapped in a deadlock. In the normal operation, users restart WDT at the regular intervals before the counter counts down to zero. WDT generates one or a combination of the following signals: Reset, interrupt, or external interrupt signals. The features of WDT include:

- 32-bit down counter
- Upon timeout, WDT outputs one or a combination of the system reset/system interrupt/external interrupt signals
- Supports variable timeout periods of reset
- Access protection

### 1.3.20 Interrupt Controller (INTC)

The interrupt controller provides both the FIQ and IRQ modes to the microprocessor. It determines the causes of the interrupts as an IRQ or an FIQ and masks the interrupts. The features of INTC include:

- Supports up to 64 fast interrupt (FIQ) inputs
- Supports up to 64 standard interrupt (IRQ) inputs
- Interrupts that can be routed to either IRQ or FIQ
- Provides both edge-triggered and level-triggered interrupt sources with positive and negative directions
- Supports de-bounce circuits for the interrupt input sources
- Independently enable or disable any interrupt sources

### 1.3.21 GPIO

GPIO is used to input or output data from or to the system and device. Each GPIO can be programmed as an input, an output, or an interrupt input. GPIO supports the rising edge, falling edge, both edges, and high-level or low-level interrupt sense type. The features of GPIO include:

- Triggers GPIO interrupt of each port
- Generates interrupt of each port triggered at rising edge, falling edge, both edges, or high-level or low–level interrupt sense type
- Pulls high or low of each port
- Provides de-bounce function of each port
- Sets or clears output data bit
- Sets to the input mode at the hardware reset for all ports

### 1.3.22 I²C Bus

The $I^2C$ bus is a two-wire bidirectional serial bus that provides a simple and efficient method for data exchange when minimizing the interconnection between devices. The $I^2C$ bus interface controller allows the host processor to serve as a master or slave residing on the $I^2C$ bus. Data are transmitted to and received from the $I^2C$ bus via a buffered interface. The features of the $I^2C$ bus include:

- Supports standard and fast modes by programming clock division register
- Supports 7-bit and 10-bit general call addressing mode
- Glitch suppression through de-bounce circuit
- Programmable slave address
- Supports master transmit, master receive, slave transmit, and slave receive modes
- General-call address detection in slave mode

### 1.3.23 Power Management Unit (PMU)

The PMU module generates the control signal in the system clock/reset/operating modes. The clocks are derived from a 32768-Hz crystal. The 32768-Hz crystal generates the 30-MHz clock of the Phase-Locked Loop (PLL) and the 30-MHz output clock drives a core PLL and a peripheral PLL to generate all system clocks. PLL can produce the desired clock frequency for particular functional blocks. The supported modes include:

- Normal mode: Normal full-function mode
- FCS mode: Frequency Change Sequence (FCS) mode allows users to change the core PLL settings so that the system can operate at different frequencies.

### 1.3.24 $I^2S$ Controller

The $I^2S$ controller is a full-duplex synchronous serial interface that can connect to various external analog-to-digital (A/D) converters as well as audio and telecom codecs. The serial data format is ranging from 4 bits to 32 bits in length. The features of the $I^2S$ controller include:

- Supports TI SSP, Motorola SPI, National Semiconductor MICROWIRE, Philips I2S
- Provides independent clock to ease bit-clock generation
- Supports master and slave modes
- Internally or externally controlled serial bit clock
- Internally or externally controlled frame/sync.
- Programmable frame/sync. polarity
- Programmable serial-bit clock polarity, phase, and frequency
- Programmable serial-bit data sequence (MSB or LSB first)
- Programmable threshold interrupt of transmit/receive FIFOs
- Independently programmable interrupt enable/disable function
- Provides 16-word transmit FIFO and 16-word receive FIFO
- Supports DMA REQ/ACK for large data transfers
- Programmable $I^2S$ format (Including zero-bit padding, right-justified, and left-justified)

### 1.3.25 UART Controller

The UART controller is a serial communications element that is backward compatible with 16550 to support the existing communication software. The features of the UART controller include:

- High-speed NS 16C550A-compatible UART
- Programmable baud rate of up to 1152 Kbps
- Ability to add or delete standard asynchronous communication bits (Start, stop, and parity) in serial data
- Programmable baud rate generator to divide the internal clock by 1 to ($2^{16}$ - 1) to generate internal 16X clock
- Fully programmable serial interface:
  - 5-bit, 6-bit, 7-bit, or 8-bit character
  - Even, odd, and no parity detections
  - 1, 1.5, or 2 stop-bit generations
- Complete status reporting capability
- Generates and detects line breaks
- Fully prioritized interrupt system controls
- Separated DMA requests for data transmission and reception services
- Break, parity, overrun, and framing error simulation in UART mode

### 1.3.26 Secure Digital Card (SDC) Controller

The SDC controller not only supports the SD and MMC interface protocols but also supports the hot insertion/removal detections. With the SD interface protocol, the SDC controller provides the write protection and 1-bit or 4-bit bus width for transferring large data without CPU intervention. The features of the SDC controller include:

- Supports SD memory card protocol, Version 1.0
- Supports SD/MMC bus protocols
- Supports 4-bit wide SD data bus
- Supports 4-word wide data FIFO
- Built-in generation and check of 7-bit and 16-bit CRC data
- Clock rate range: 0 MHz ~ 50 MHz for memory card
- Hot insertion/removal detection

- Write-protect SD card
- Supports single-block or multiple-block accesses for read, write, and erase operations
- Supports DMA REQ/ACK for large data transfers

### 1.3.27 Comparsion Table of GM8125/GM8126/GM8128

| Product Aias | P/N | Package Type | Max Encode Performance | ISP |
|---|---|---|---|---|
| GM8128 | GM8128SS-QC | E-PAD TQFP-176-20x20 | 1080p @ 30 fps | V |
| GM8128 | GM8128PS-BC | TFBGA-256-12x12 | 1080p @ 30 fps | V |
| GM8126 | GM8126SF-QC | E-PAD TQFP-176-20x20 | 720p @ 45 fps | V |
| GM8126 | GM8126PF-BC | TFBGA-256-12x12 | 720p @ 45 fps | V |
| GM8125 | GM8125EL-QC | E-PAD TQFP-176-20x20 | 720p @ 45 fps | N/A |

"V": indicates "Available".

### 1.3.28 Comparsion Table of E-PAD TQFP-176 and TFBGA-256

| Component | E-PAD TQFP-176 | TFBGA-256 |
|---|---|---|
| FA626 | Y | Y |
| H.264 encoder | Y | Y |
| MPEG4 codec | Y | Y |
| JPEG codec | Y | Y |
| ISP | Y | Y |
| Video captrue | Y | Y |
| 3DDI | Y | Y |
| AES/DES | Y | Y |
| DMA | Y | Y |
| LCDC | Y | Y |
| USB 2.0 OTG | Y | Y |
| 3.3 V to 2.5 V regulator | Y | Y |
| 10/100M Ethernet PHY | Y | Y |
| Audio codec | Y | Y |
| SAC ADC | Y | Y |
| NTSC/PAL composite output | Y | Y |
| NAND boot | N | Y |

| Component | E-PAD TQFP-176 | TFBGA-256 |
|---|---|---|
| SPI boot | Y | Y |
| NAND Flash I/F | N | Y |
| DDR2 I/F | Y | Y |
| SD card I/F | Y | Y |
| 12-bit Bayer RGB I/F | Y | Y |
| BT.656/BT.601 input | Y | Y |
| BT.656 output | Y | Y |
| BT.1120 input | Y | Y |
| BT.1120 output | N | Y |
| RMII/MII I/F | N | Y |
| PWM | Y | Y |
| Timer | Y | Y |
| WDT | Y | Y |
| INTC | Y | Y |
| RTC | Y | Y |
| I2C | Y | Y |
| SPI | Y | Y |
| I2S | Y | Y |
| UART0 | Y | Y |
| UART1 | Y | Y |
| UART2 | N | Y |
| UART3 | N | Y |
| UART4 | N | Y |
| GPIO | Y | Y |

Notes 1: "Y" indicates "Support"

2: "N" indicates " Do not support"

# Chapter 2

# Signal Description

This chapter contains the following sections:

- 2.1 Signal Description
- 2.2 TFBGA-256 Pin Assignments

## 2.1 Signal Description

Table 2-1 lists and describes the signals of GM8126. In the "Direction" column, "I" indicates the input signals, "O" indicates the output signals, "I/O" indicates the bidirectional signals, "A" indicates the analog signals, "SUP" indicates the power supply pins, "PU" indicates the internal pull-up input or bidirectional signals, "PD" indicates the internal pull-down input or bidirectional signals, and "PP" indicates the programmable internal pull-up or pull-down input or bidirectional signals.

**Table 2-1. TFBGA-256 Signal Descriptions**

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| **FA626 ICE signals** | | | |
| NTRST | I | F3 | ICE Reset<br>Active low |
| TCK | I | F2 | ICE clock |
| TMS | I | E4 | ICE mode select |
| TDO | O | F4 | ICE data output |
| TDI | I | F1 | ICE data input |
| **CAPTURE interface signals** | | | |
| CAP0_FIELD | I | G4 | CAPTURE0 field input |
| CAP0_HS | I | G1 | CAPTURE0 HS input |
| CAP0_VS | I | G2 | CAPTURE0 VS input |
| CAP0_D[7:0] | I | G3, H4, H3, H2, H1, J4, J3, J2 | CAPTURE0 data input |
| CAP0_CLK | I | K4 | CAPTURE0 pixel clock input |
| CAP_CLKOUT | O | J1 | CAPTURE clock output |
| CAP1_CLK | I | K3 | CAPTURE1 pixel clock input |
| CAP1_D[7:0] | I | K2, K1, L4, L3, L2, L1, M2, M1 | CAPTURE1 data input |
| VCCIO_CAP | SUP | J5, K5 | Positive supply for all I/O pins of capture interface.<br>This signal must be connected from the 1.8-V voltage supply to the 3.3-V voltage supply on PCB. |
| **DDR2 DRAM interface signals** | | | |
| CK,CKB | O | C16, B16 | Differential clock outputs |
| A[13:0] | O | A13, A10, B13, A11, D11, C14, B10, D14, B11,B14, C11, A14, D12, D15 | Address bus |
| BA[2:0] | O | C12, B12, A12 | Bank selection |
| DQ[15:0] | I/O | G13, K13, F14, J13, K16, F13, K14, G14, E14, J14, E16, H13, H14, E15, J16, E13 | Data bus |
| DQS[1:0] | I/O | H16, F16 | Data strobe |

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| DQSB[1:0] | I/O | H15, F15 | Used in the differential DQS mode |
| CS | O | B15 | Chip select |
| RAS | O | A15 | Row address strobe |
| CAS | O | C15 | Column address strobe |
| WE | O | D13 | Write enable |
| DQM[1:0] | O | K15, J15 | Data mask |
| CKE | O | C13 | Clock enable |
| ODT | O | A16 | On-Die Termination (ODT) control |
| VREF | SUP | D16 | Reference voltage for receivers |
| **Composite output signals** | | | |
| TVE_COMP | O | N6 | Bandwidth/noise reduction node |
| TVE_RSET | O | T6 | Full-scale output current adjustment |
| TVE_IOUT | O | T7 | Positive current output |
| VCC33A_TVE | SUP | P7 | 3.3 V analog supply voltage |
| GND33A_TVE | SUP | R7 | Analog ground supply |
| **Ethernet PHY Interface signals** | | | |
| EDP_TXOP | O | N15 | TX differential signal pair |
| EDP_TXON | O | N16 | TX differential signal pair |
| EDP_RXIP | I | P15 | RX differential signal pair |
| EDP_RXIN | I | P16 | RX differential signal pair |
| EDP_RSET | O | T16 | Connect a resistor of 12.3 kΩ ±1% to the ground |
| EDP_LINKLED | O | P14 | Link status LED |
| EDP_SPDLED | O | N14 | Link speed LED |
| VCC25A_EDP | SUP | M14, R15 | 2.5-V analog supply voltage |
| GNDA_EDP | SUP | R16, T15 | 2.5-V analog ground supply |
| **Regulator interface signals** | | | |
| REG_VO25 | SUP | M16 | Regulator output 2.5-V supply voltage |
| VCC33_REG | SUP | M15 | Regulator input 3.3-V supply voltage |
| **USB 2.0 OTG signals** | | | |
| OTG_DP | I/O | R6 | Differential data in positive terminal |
| OTG_DM | I/O | T5 | Differential data in negative terminal |
| OTG_RREF | I | P6 | This signal connects the external reference resistor to the analog GND |
| VCC33A_OTG | SUP | R5 | 3.3-V analog supply voltage |
| GND33A_OTG | SUP | P5 | Analog ground supply |

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| **ADC interface signals** | | | |
| AIN0 | A | T4 | ADC channel 0 input |
| AIN1 | A | N5 | ADC channel 1 input |
| **Audio codec interface signals** | | | |
| SPKOUTP | A | T2 | Analog speaker output of the positive channel |
| SPKOUTN | A | T1 | Analog speaker output of the negative channel |
| LMIC | A | R3 | Analog microphone input of the left channel |
| RMIC | A | T3 | Analog microphone input of the right channel |
| AUDIO_VCM | A | R4 | Analog common-mode voltage |
| VCC33A_AUDIO | SUP | P4 | Analog 3.3-V supply voltage |
| GND33A_AUDIO | SUP | P3 | Analog 3.3-V ground supply |
| VCC33A_SPK | SUP | R2 | Analog 3.3-V supply voltage |
| GND33A_SPK | SUP | R1 | Analog 3.3-V ground supply |
| **TV digital data output interface signals** | | | |
| TV[15:0] | O | E2, E3, E1, D4, D3, D2, D1, A1, C2, B2, A2, A3, C4, B4, A4, D5 | TV digital data output |
| **SD card interface signals** | | | |
| SD_CD | I | N7 | SD card detect signal derived from the connector<br>This signal is not an SD bus standard signal. |
| SD_CLK | O | T9 | SD card clock signal |
| SD_CMD_RSP | I/O | R9 | SD card command/response signal |
| **NAND Flash interface signals** | | | |
| NAND_CEN0 | O | T12 | NAND flash chip enable |
| NAND_CLE | O | P12 | NAND flash chip command latch |
| NAND_ALE | O | T13 | NAND flash chip address latch |
| NAND_WEN | O | R13 | NAND flash chip write enable |
| NAND_REN | O | R12 | NAND flash chip read enable |
| NAND_BUSYN | I | N11 | NAND flash chip busy |
| NAND_D[3:0] | I/O | N9, P9, N8, P8 | NAND flash chip data[3:0] bus |
| SD_D[3:0] | I/O | | SD card data bus |
| NAND_D[7:4] | I/O | T14, R14, P13, N13 | NAND flash chip data[7:4] bus |

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| **I²C interface signals** | | | |
| I2C_SCL | I/O | P1 | I²C clock |
| I2C_SDA | I/O | P2 | I²C data |
| **I²S interface signals** | | | |
| I2S_FS | I/O | M3 | I²S frame sync0 |
| I2S_SCLK | I/O | N3 | I²S bit clock |
| I2S_TXD | O | N4 | I²S data-out |
| I2S_RXD | I | M4 | I²S data-in |
| **SPI interface signals** | | | |
| SPI_FS0 | O | R8 | SPI frame sync0 |
| SPI_FS1 | O | T8 | SPI frame sync1 |
| SPI_SCLK | O | R10 | SPI bit-clock |
| SPI_TXD | O | P10 | SPI data-out |
| SPI_RXD | I | T10 | SPI data-in |
| **UART 0 signals** | | | |
| UART0_SIN | I | R11 | UART 0 receive |
| UART0_SOUT | O | T11 | UART 0 transmit |
| **UART 1 signals** | | | |
| UART1_SIN | I | N10 | UART 1 receive |
| UART1_SOUT | O | P11 | UART 1 transmit |
| **UART 2 signals** | | | |
| UART2_SIN | I | N11 | UART 2 receive |
| UART2_SOUT | O | N12 | UART 2 transmit |
| **UART 3 signals** | | | |
| UART3_SIN | I | B6 | UART 3 receive |
| UART3_SOUT | O | A5 | UART 3 transmit |
| **UART 4 signals** | | | |
| UART4_SIN | I | C6 | UART 4 receive |
| UART4_SOUT | O | D6 | UART 4 transmit |
| **GPIO signals** | | | |
| PWM[3:0] | O | C5, B5, N2, N1 | General-purpose I/O |
| GPIO[3:0] | I/O | | |
| GPIO2[31:28] | I/O | C1, B1, C3, B3 | General-purpose I/O |
| **PMU signals** | | | |
| OSCLI | A | B8 | 32768-Hz crystal input |
| OSCLO | A | A8 | 32768-Hz crystal output |

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| PWREN | O | C10 | Power enable output |
| WAKEUP | I | A9 | Wakeup enable input |
| BATFULTN | I | B9 | Battery fault input |
| RSTN | I | A6 | Hardware reset input |
| VCC3A_RTC | SUP | C9 | 3.3 V voltage supply |
| OM | I | E9 | Test mode input |
| **OSCH signals (Optional)** | | | |
| OSCHI | A | B7 | 30-MHz crystal input |
| OSCHO | A | A7 | 30-MHz crystal output |
| VCCA_OSC | SUP | D7 | Positive supply of the oscillator<br>This signal must be connected to the 3.3-V voltage supply. |
| GND33A_OSC | SUP | C7 | Grounded supply of the oscillator |
| **PLL power/ground** | | | |
| VCC10A_PLL | A | D8 | 1.0 V analog power supply |
| GND10A_PLL | A | C8 | 1.0 V analog ground |
| PLL_LPF | I | D9 | External loop filter for PLL |
| **DLL power/ground** | | | |
| VCC10A_DLL | A | G16 | 1.0 V analog power supply |
| GND10A_DLL | A | G15 | 1.0 V analog ground |
| **Powers/Grounds** | | | |
| VCC3IO | SUP | E5, F5, M5, M12, M13 | Positive supply of all I/O pins, except DDR<br>This signal must be connected to the 3.3-V voltage supply on PCB. |
| VCC10 | SUP | E6, E7, F6, G6, M7, M8, M9, M10, F10, F11, G10, G11 | Positive supply of the internal logic<br>This signal must be connected to the 1.0-V voltage supply on PCB. |
| VCC18 | SUP | E11, E12, F12, K12, L12, L13, L14 | Positive supply of the DDR I/O pins<br>This signal must be connected to the 1.8-V voltage supply on PCB. |
| GND_DDR | SUP | D10, E10, G12, H12, J12, L15, L16 | Ground supply of the DDR I/O pins |
| GND | SUP | E8, F7, F8, F9, G5, G7, G8, G9, H5, H6, H7, H8, H8, H10, H11, J6, J7, J8, J9, J10, J11,<br>K6, K7, K8, K9, K10,K11, L5, L6, L7, L8, L9, L10, L11, M6, M11 | Ground supply of all the I/O pins and internal logic |

**Table 2-2. E-PAD TQFP-176 Signal Descriptions**

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| **FA626 ICE signals** | | | |
| NTRST | I | 2 | ICE reset<br>Active low |
| TCK | I | 3 | ICE clock |
| TMS | I | 4 | ICE mode select |
| TDO | O | 5 | ICE data output |
| TDI | I | 6 | ICE data input |
| **CAPTURE interface signals** | | | |
| CAP0_FIELD | I | 8 | CAPTURE0 field input |
| CAP0_HS | I | 9 | CAPTURE0 HS input |
| CAP0_VS | I | 10 | CAPTURE0 VS input |
| CAP0_D[7:0] | I | 11, 12, 13, 14, 15, 16, 17, 18 | CAPTURE0 data input |
| CAP0_CLK | I | 19 | CAPTURE0 pixel clock input |
| CAP_CLKOUT | O | 21 | CAPTURE clock output |
| CAP1_CLK | I | 22 | CAPTURE1 pixel clock input |
| CAP1_D[7:0] | I | 23, 24, 25, 26, 27, 28, 29, 30 | CAPTURE1 data input |
| **DDR2 DRAM interface signals** | | | |
| CK, CKB | O | 134,135 | Differential clock outputs |
| A[13:0] | O | 147, 161, 148, 156, 160, 145, 162, 144, 157, 143, 158, 142, 154, 141 | Address bus |
| BA[2:0] | O | 153, 152, 151, | Bank selection |
| DQ[15:0] | I/O | 120, 102, 123, 106, 105, 122, 104, 121, 126, 109, 132, 112, 113, 130, 111, 129 | Data bus |
| DQS[1:0] | I/O | 115, 125 | Data strobe |
| DQSB[1:0] | I/O | 114, 124 | Used in the differential DQS mode |
| CS | O | 139 | Chip select |
| RAS | O | 138 | Row address strobe |
| CAS | O | 140 | Column address strobe |
| WE | O | 150 | Write enable |
| DQM[1:0] | O | 103, 110 | Data mask |
| CKE | O | 149 | Clock enable |
| ODT | O | 136 | On-Die Termination (ODT) control |
| VREF | SUP | 100 | Reference voltage for receivers |

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| **Composite output signals** | | | |
| TVE_COMP | O | 61 | Bandwidth/Noise reduction node |
| TVE_RSET | O | 62 | Full-scale output current adjustment |
| TVE_IOUT | O | 65 | Positive current output |
| VCC33A_TVE | SUP | 63 | 3.3-V analog supply voltage |
| GND33A_TVE | SUP | 64 | Analog ground supply |
| **Ethernet PHY interface signals** | | | |
| EDP_TXOP | O | 97 | TX differential signal pair |
| EDP_TXON | O | 98 | TX differential signal pair |
| EDP_RXIP | I | 94 | RX differential signal pair |
| EDP_RXIN | I | 95 | RX differential signal pair |
| EDP_RSET | O | 92 | Connect a resistor of 12.3 kΩ ±1% to the ground |
| EDP_LINKLED | O | 80 | Link status LED |
| EDP_SPDLED | O | 81 | Link speed LED |
| VCC25A_EDP | SUP | 91, 96 | 2.5-V analog supply voltage |
| GNDA_EDP | SUP | 93, 99 | 2.5-V analog ground supply |
| **Regulator interface signals** | | | |
| REG_VO25 | SUP | 90 | Regulator output 2.5-V supply voltage |
| VCC33_REG | SUP | 89 | Regulator input 3.3-V supply voltage |
| **USB 2.0 OTG signals** | | | |
| OTG_DP | I/O | 57 | Differential data in positive terminal |
| OTG_DM | I/O | 56 | Differential data in negative terminal |
| OTG_RREF | I | 59 | This signal connects the external reference resistor to the analog GND |
| VCC33A_OTG | SUP | 55 | 3.3-V analog supply voltage |
| GND33A_OTG | SUP | 58 | Analog ground supply |
| **ADC interface signals** | | | |
| AIN0 | A | 53 | ADC channel 0 input |
| AIN1 | A | 52 | ADC channel 1 input |
| **Audio codec interface signals** | | | |
| SPKOUTP | A | 44 | Analog speaker output of the positive channel |
| SPKOUTN | A | 43 | Analog speaker output of the negative channel |
| RLINEIN | A | 50 | Analog line input of the right channel |
| LLINEIN | A | 49 | Analog line input of the left channel |
| LMIC | A | 51 | Analog microphone input |
| AUDIO_VCM | A | 47 | Analog common-mode voltage |

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| VCC33A_AUDIO | SUP | 48 | Analog 3.3-V supply voltage |
| GND33A_AUDIO | SUP | 46 | Analog 3.3-V ground supply |
| VCC33A_SPK | SUP | 45 | Analog 3.3-V supply voltage |
| GND33A_SPK | SUP | 42 | Analog 3.3-V ground supply |
| **SD card interface signals** | | | |
| SD_CD | I | 68 | SD card detect signal derived from the connector<br>This signal is not an SD bus standard signal. |
| SD_CLK | O | 69 | SD card clock signal |
| SD_CMD_RSP | I/O | 70 | SD card command/response signal |
| SD_D[3:0] | I/O | 74, 73, 72, 71 | SD card data bus |
| **I²C interface signals** | | | |
| I2C_SCL | I/O | 38 | I²C clock |
| I2C_SDA | I/O | 39 | I²C data |
| **I²S interface signals** | | | |
| I2S_FS | I/O | 32 | I²S frame sync0 |
| I2S_SCLK | I/O | 34 | I²S bit clock |
| I2S_TXD | O | 35 | I²S data-out |
| I2S_RXD | I | 33 | I²S data-in |
| **SPI interface signals** | | | |
| SPI_FS0 | O | 75 | SPI frame sync0 |
| SPI_FS1 | O | 76 | SPI frame sync1 |
| SPI_SCLK | O | 78 | SPI bit-clock |
| SPI_TXD | O | 79 | SPI data-out |
| SPI_RXD | I | 77 | SPI data-in |
| **UART 0 signals** | | | |
| UART0_SIN | I | 87 | UART 0 receive |
| UART0_SOUT | O | 88 | UART 0 transmit |
| **UART 1 signals** | | | |
| UART1_SIN | I | 82 | UART 1 receive |
| UART1_SOUT | O | 83 | UART 1 transmit |
| **UART 2 signals** | | | |
| UART2_SOUT | O | 84 | UART 2 transmit |
| **GPIO signals** | | | |
| PWM[3:0] | O | 175, 176, 36, 37 | PWM output |
| GPIO[3:0] | I/O | | General-purpose I/O |

| Signal Name | Direction | Pin Number | Description |
|---|---|---|---|
| **PMU signals** | | | |
| OSCLI | A | 168 | 32768-Hz crystal input |
| OSCLO | A | 169 | 32768-Hz crystal output |
| PWREN | O | 166 | Power enable output |
| WAKEUP | I | 167 | Wakeup enable input |
| VCC3A_RTC | SUP | 171 | 3.3-V voltage supply |
| OM | I | 170 | Test mode input |
| **OSCH signals (Optional)** | | | |
| OSCHI | A | 173 | 30-MHz crystal input |
| OSCHO | A | 174 | 30-MHz crystal output |
| VCCA_OSC | SUP | 172 | Positive supply of the oscillator<br>This signal must be connected to the 3.3-V voltage supply. |
| **PLL powers/grounds** | | | |
| VCC10A_PLL | A | 163 | 1.0-V analog power supply |
| GND10A_PLL | A | 164 | 1.0-V analog ground |
| PLL_LPF | I | 165 | External loop filter for PLL |
| **DLL powers/grounds** | | | |
| VCC10A_DLL | A | 118 | 1.0-V analog power supply |
| GND10A_DLL | A | 117 | 1.0-V analog ground |
| **Powers/Grounds** | | | |
| VCC3IO | SUP | 7, 41, 67, 86 | Positive supply of all I/O pins, except DDR<br>This signal must be connected to the 3.3-V voltage supply on PCB. |
| VCCIO_VCAP | SUP | 20, 31 | Positive supply of all I/O pins, except DDR<br>This signal must be connected to the 3.3-V voltage supply on PCB. |
| VCC10 | SUP | 1,40, 54, 60, 66, 85, 107, 127, 137, 159 | Positive supply of the internal logic<br>This signal must be connected to the 1.0-V voltage supply on PCB. |
| VCC18 | SUP | 101, 108, 116, 119, 128, 131, 133, 146,155 | Positive supply of the DDR I/O pins<br>This signal must be connected to the 1.8-V voltage supply on PCB. |

**Table 2-3. Jumper Settings**

| Function | jpset | Pin Name | Description | Default Value |
|---|---|---|---|---|
| PLL1 output | jpset[0] | SSP0_FS0 | PLL1 output frequency setting<br>0: PLL1 output frequency is 800 MHz.<br>1: PLL1 output frequency is 400 MHz. | 1 |
| PLL 32k disable | jpset[1] | SSP0_FS1 | 0: Disable pll32k<br>1: Enable pll32k | 1 |
| Test mode | jpset[2] | UART1_SOUT | Reserved | 1 |
| Test mode | jpset[3] | NAND_DATA[5] | Reserved | 1 |
| Test mode | jpset[4] | UART2_SOUT | Reserved | 1 |
| NAND boot | jpset[5] | NAND_DATA[7] | 0: SPI boot<br>1: NAND boot | 1 |
| Test mode | jpset[6] | EDP_SPDLED | Reserved | 1 |
| Firmware update | jpset[7] | EDP_LINKLED | 0: Firmware update mode<br>1: Normal mode | 1 |

## 2.2 TFBGA-256 Pin Assignments

Figure 2-1 shows the pin assignments of Ball Grid Array (BGA) for GM8126.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | TV_D8 | TV_D5 | TV_D4 | TV_D1 | UART3_SOUT | RSTN | OSCHO | OSCLO | WAKEUP | A12 | A10 | BA0 | A13 | A2 | RAS | ODT | A |
| B | GPIO2[30] | TV_D6 | GPIO2[28] | TV_D2 | PWM2 | UART3_SIN | OSCHI | OSCLI | BATFAULTN | A7 | A5 | BA1 | A11 | A4 | CS | CKB | B |
| C | GPIO2[31] | TV_D7 | GPIO2[29] | TV_D3 | PWM3 | UART4_SIN | GND33A_OSC | GND10A_PLL | VCC3A_RTC | PWREN | A3 | BA2 | CKE | A8 | CAS | CK | C |
| D | TV_D9 | TV_D10 | TV_D11 | TV_D12 | TV_D0 | UART4_SOUT | VCCA_OSC | VCC10A_PLL | PLL_LPF | GND_DDR | A9 | A1 | WE | A6 | A0 | VREF_DDR | D |
| E | TV_D13 | TV_D15 | TV_D14 | TMS | VCC3IO | VCCK | VCCK | GND | OM | GND_DDR | VCC_DDR | VCC_DDR | DQ0 | DQ7 | DQ2 | DQ5 | E |
| F | TDI | TCK | NTRST | TDO | VCC3IO | VCCK | GND | GND | GND | VCCK | VCCK | VCC_DDR | DQ10 | DQ13 | DQSB0 | DQS0 | F |
| G | CAP0_HS | CAP0_VS | CAP0_D7 | CAP0_FIELD | GND | VCCK | GND | GND | GND | VCCK | VCCK | GND_DDR | DQ15 | DQ8 | GND10A_DLL | VCC10A_DLL | G |
| H | CAP0_D3 | CAP0_D4 | CAP0_D5 | CAP0_D6 | GND | GND | GND | GND | GND | GND | GND | GND_DDR | DQ4 | DQ3 | DQSB1 | DQS1 | H |
| J | CAP_CLKOUT | CAP0_D0 | CAP0_D1 | CAP0_D2 | VCCIO_CAP | GND | GND | GND | GND | GND | GND | GND_DDR | DQ12 | DQ6 | DQM0 | DQ1 | J |
| K | CAP1_D6 | CAP1_D7 | CAP1_CLK | CAP0_CLK | VCCIO_CAP | GND | GND | GND | GND | GND | GND | VCC_DDR | DQ14 | DQ9 | DQM1 | DQ11 | K |
| L | CAP1_D2 | CAP1_D3 | CAP1_D4 | CAP1_D5 | GND | GND | GND | GND | GND | GND | GND | VCC_DDR | VCC_DDR | VCC_DDR | GND_DDR | GND_DDR | L |
| M | CAP1_D0 | CAP1_D1 | I2S_FS | I2S_RXD | VCC3IO | GND | VCCK | VCCK | VCCK | VCCK | GND | VCC3IO | VCC3IO | VCC25A_EDP | VCC33_REG | REG_V025 | M |
| N | PWM0 | PWM1 | I2S_SCLK | I2S_TXD | AIN1 | TVE_COMP | SD_CD | NAND_D1 | NAND_D3 | UART1_SIN | UART2_SIN | UART2_SOUT | NAND_D4 | EDP_SPDLED | EDP_TXOP | EDP_TXON | N |
| P | I2C_SCL | I2C_SDA | GND33A_AUDIO | VCC33A_AUDIO | GND33A_OTG | OTG_RREF | VCC33A_TVE | NAND_D0 | NAND_D2 | SPI_TXD | UART1_SOUT | NAND_CLE | NAND_D5 | EDP_LINKLED | EDP_RXIP | EDP_RXIN | P |
| R | GND33A_SPK | VCC33A_SPK | LMIC | AUDIO_VCM | VCC33A_OTG | OTG_DP | GND33A_TVE | SPI_FS0 | SD_CMD | SPI_SCLK | UART0_SIN | NAND_REN | NAND_WEN | NAND_D6 | VCC25A_EDP | GNDA_EDP | R |
| T | SPKOUTN | SPKOUTP | RMIC | AIN0 | OTG_DM | TVE_RSET | TVE_IOUT | SPI_FS1 | SD_CLK | SPI_RXD | UART0_SOUT | NAND_CEN0 | NAND_ALE | NAND_D7 | GNDA_EDP | EDP_RSET | T |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |

**Figure 2-1. TFBGA-256 Pin Assignments (Top View)**

# Chapter 3

# Memory Map

This chapter contains the following section:

- 3.1 Memory Map

## 3.1 Memory Map

GM8126 with a 32-bit address bus is capable of addressing a 4-GB physical address space. This space is divided into the section regions within which the various memories and peripherals are mapped. Figure 3-1 shows the detailed memory map of GM8126.

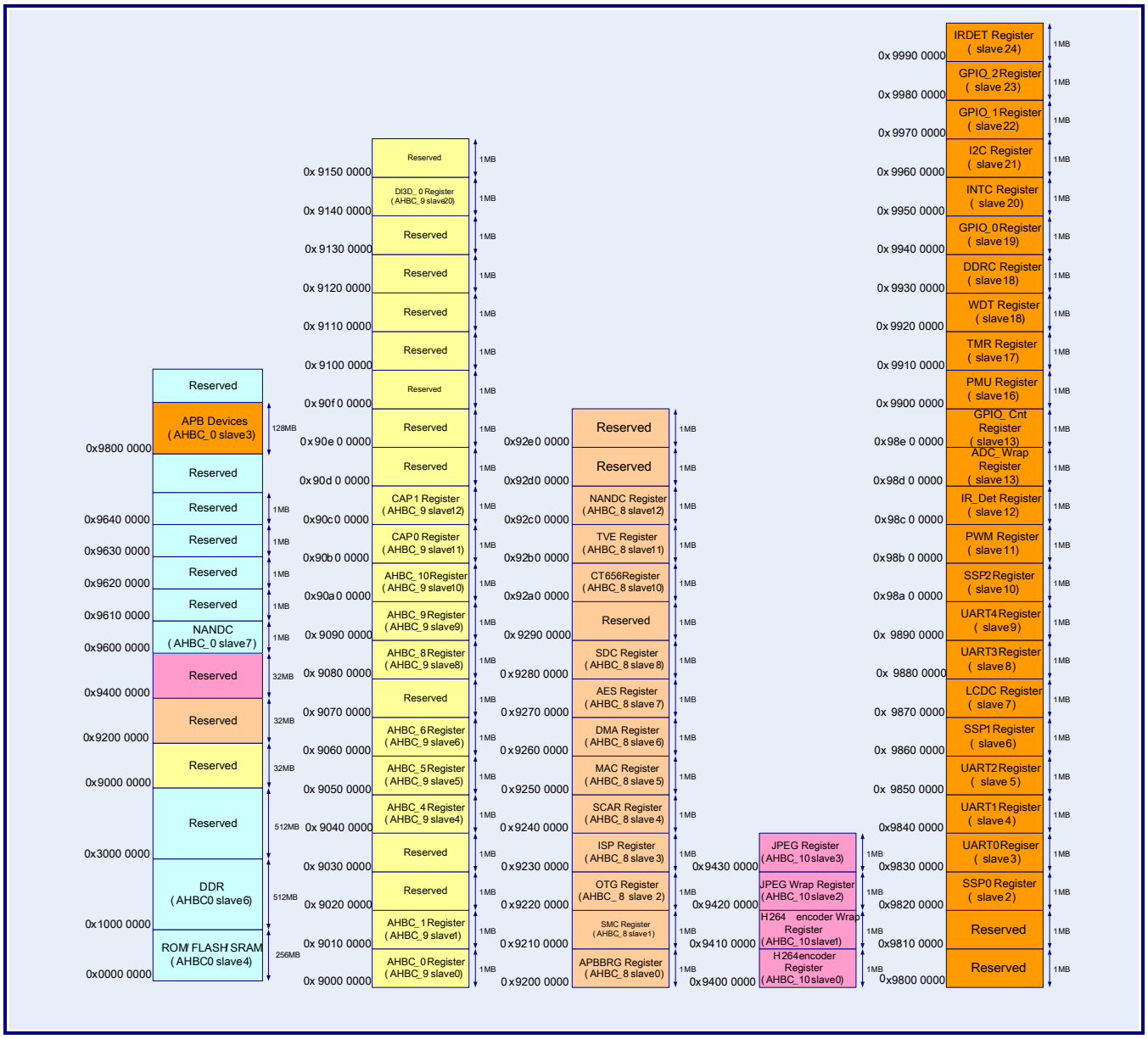



**Figure 3-1. Physical Memory Map of GM8126**

# Chapter 4

# Power Management Unit

This chapter contains the following sections:


- 4.1 General Description
- 4.2 Features
- 4.3 Clock Manager
- 4.4 Programming Model

## 4.1 General Description

The Power Management Unit (PMU) provides fixed clocks for each peripheral unit. Many peripheral clocks of the devices can be disabled by using the AHB Module Clock Off Control Register (AHBMCLKOFF), the APB Module Clock Off Control Register (APBMCLKOFF), or through the bits in the peripheral control registers. The clock to any unit that is not used will be turned off to minimize the power consumption. PMU also provides a method to change the PLL frequency.

## 4.2 Features

- Supports hardware reset (Asserted RSTN, only for TFBGA-256) for a total non-maskable reset
- Supports watchdog reset to assert watchdog timer for resetting system, except for PMU units, and can be used as a code monitor
- Supports normal mode to enable all power supplies and clocks
- Supports FCS mode to change settings of core PLL for operating at various system frequencies

## 4.3 Clock Manager

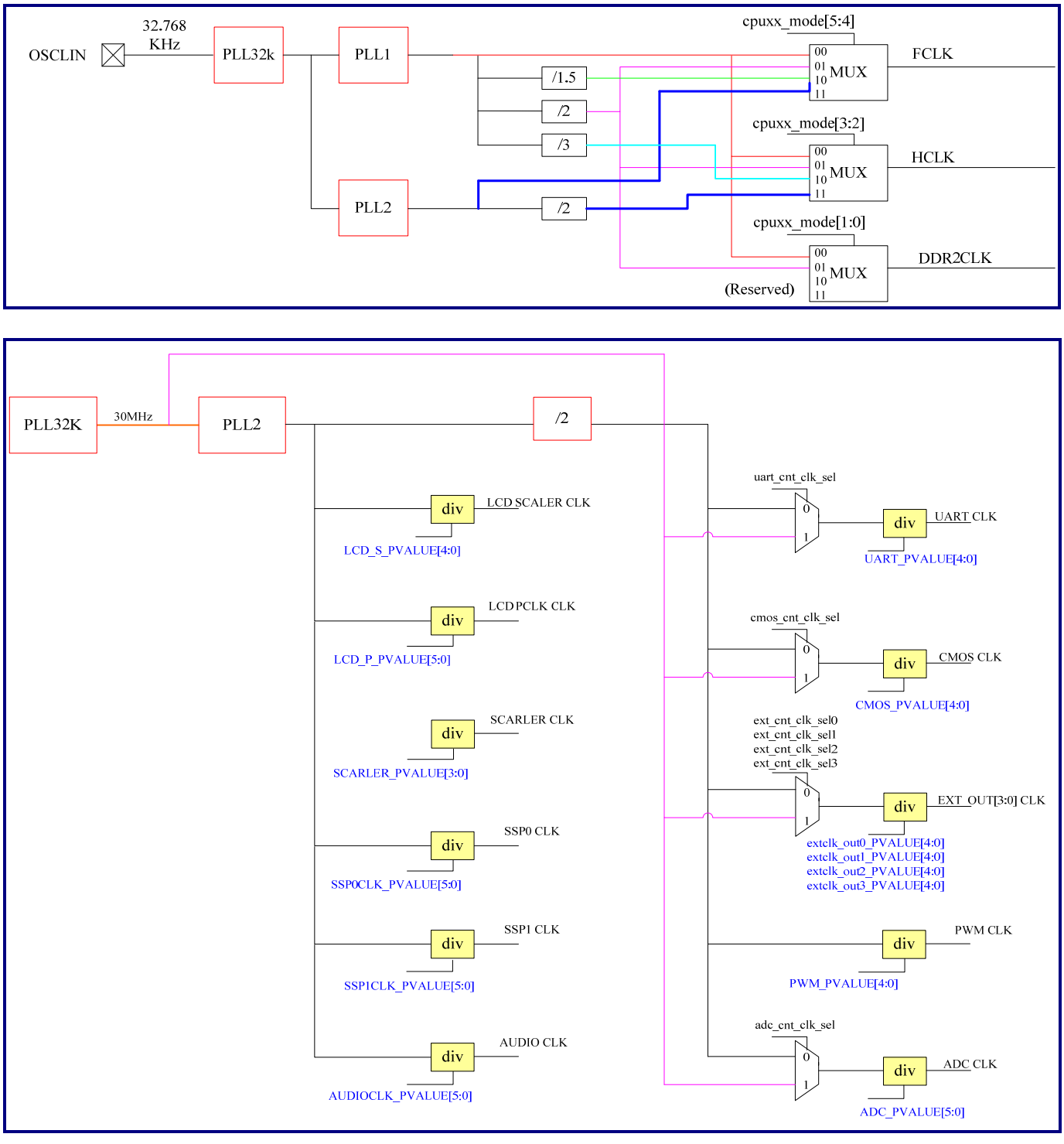



**Figure 4-1. PMU Block Diagram for PLL32K, PLL1, and PLL2**

The GM8126 clocking system incorporates the following clock sources:

- 32.768-kHz or 30-MHz oscillator
- 30-MHz clock frequency output (PLL32k)
- Programmable frequency of the core PLL (PLL1)
- Programmable frequency of the peripheral PLL (PLL2)

### 4.3.1 32.768-kHz Oscillator

The 32.768-kHz oscillator is the primary clock source of GM8126. The on-chip PLL32k frequency multiplier uses the 32.768-kHz oscillator as a reference to generate 30-MHz for PLL1/PLL2 as the clock input source.

### 4.3.2 Core PLL (PLL1)

PLL1 is the clock source of the AHB bus devices, APB bus devices, FA626 CPU, and DDR2 controller. PLL1 uses the 30-MHz oscillator as a reference and its frequency is controlled by programming the PLL1NS[8:0] and PLL1MS[4:0] bits in the PLL1CR register. The following equation shows the calculation of the PLL1 frequency output.

**PLL1 frequency output = (30 * PLL1NS/PLL1MS) MHz**

The AHB bus clock (HCLK) is the calculation result of dividing the PLL1 frequency by 1, 2, 3, or PLL2/2. The APB bus clock is the calculation result of dividing the AHB bus frequency by 2. The FA626 CPU bus clock (FCLK) is the calculation result of dividing the PLL1 frequency by 1, 2, 1.5, or PLL2. The DDR2 clock is the calculation result of dividing the PLL1 frequency by 1 or 2. Table 4-1 shows the relationship between the FCLK/HCLK/DDR2CLK values and the PLL1 frequency.

**Table 4-1. Relationship between FCLK/HCLK/DDR2CLK and PLL1**

| cpu_modexx[5:4] | FCLK | cpuxx_mode[3:2] | HCLK | cpuxx_mode[1:0] | DDR2CLK |
|---|---|---|---|---|---|
| 00 | PLL1/1 | 00 | PLL1/1 | 00 | PLL1/1 |
| 01 | PLL1/2 | 01 | PLL1/2 | 01 | PLL1/2 |
| 10 | PLL1/1.5 | 10 | PLL1/3 | 10 | X |
| 11 | PLL2 | 11 | PLL2/2 | 11 | X |

The output frequency of PLL1 can also be set using the jumper setting pins through the SPI_FS0 pin after the hardware reset. Table 4-2 shows the setting of the frequency output of PLL1.

**Table 4-2. Jumper Setting for Frequency Output of PLL1**

| jpset[0] | PLL1 | FCLK | HCLK | DDR2CLK |
|---|---|---|---|---|
| 1 | 400 MHz | 200 MHz | 200 MHz | 400 MHz |
| 0 | 800 MHz | 533 MHz | 266 MHz | 800 MHz |

Note: The unit is MHz.

### 4.3.3 Peripheral PLL (PLL2)

PLL2 generates the clock sources to be used by UART, $I^2S$, and PWM. PLL2 uses a 30-MHz oscillator as a reference to generate higher frequencies by programming the PLL2NS[8:0] and PLL2MS[4:0] bits in the PLL2 CR register. The following equation shows the calculation of the PLL2 frequency output.

**PLL2 frequency output = (30 * PLL2NS/PLL2MS) MHz**

The default value of the PLL2 output frequency is 540 MHz.

### 4.3.4 Clock Gating

PMU contains the AHBMCLKOFF and APBMCLKOFF registers. These registers contain the configuration bits that can disable the clocks to individual units. The configuration bits are used when a module is not used. After the hardware reset, any module that is not used must have its clock disabled.

### 4.3.5 Hardware Reset

The hardware reset is invoked when the RSTN pin is pulled low by an external source. GM8126 does not provide the method for masking or disabling the propagation of the external pin value. When the RSTN pin is asserted, the hardware reset will be invoked regardless of the operation mode.

### 4.3.6 Watchdog Reset

The watchdog reset is invoked when the software fails to properly prevent the watchdog timeout event. The watchdog reset is assumed to be generated only when the destroyed data are not properly executed by software.

### 4.3.7 Normal Mode

All power supplies are enabled and all clocks that are functionally enabled are running. The normal mode is entered subsequent to any power mode, power sequence, or after a reset completes its sequence.

- Entering condition: (Chip initial → Normal mode)
  The normal mode is entered after the system power-on or after a reset is complete.

- Exiting condition: (Normal mode → IDLE mode, Standby mode, or Sleep mode)
  The normal mode is exited when the IDLE mode is executed, or when a reset begins. For the detailed sequence, please refer to the IDLE mode.

- State definition: The normal mode is the normal operating mode of the application processor. In the normal mode, the software can use the AHBMCLKOFF (Offset = 0x38) and APBMCLKOFF (Offset = 0x3c) registers to turn on or turn off the related module clock.

### 4.3.8 IDLE Mode

- Entering condition: (Normal mode → IDLE mode)
  The software sets bits[1:0] of the Power Mode Register to 2'b01, then writes the coprocessor instruction to CPU as below.
  __asm MCR P15, 0, temp, c7, c0, 4.

- Exiting condition: (IDLE mode → Normal mode)
  The program will exit from the IDLE mode when any reset or IRQ/FIQ interrupt is asserted by INTC.

- State definition (In the IDLE mode):
  - Optionally stops the CPU clocks.
  - The remaining application processors operate normally. For example, the LCD controller can continue refreshing the screen with the same frame buffer data in the memory.
  - Interrupts are recognized as the wake-up sources.

### 4.3.9 Standby Mode

- Entering condition: (Normal mode → Standby mode)
  The software sets bits[1:0] of the Power Mode Register to 2'b10, then writes the coprocessor instruction to CPU as below.
  __asm MCR P15, 0, temp, c7, c0, 4.

- Exiting condition: (Standby mode → Normal mode)
  The program will exit from the standby mode when the external wakeup pin is activated or any reset or IRQ/FIQ interrupt is asserted by INTC.

- State definition (In the standby mode):
  - All clocks stop.
  - PLL stops.
  - Can optionally stop OSCH.
  - DRAM in self-refresh mode.
  - Interrupts are recognized as the wake-up sources.

### 4.3.10 Sleep Mode

- Entering condition: (Normal mode → Sleep mode)
  The software sets bits[1:0] of the Power Mode Register to 2'b11, then writes the coprocessor instruction to CPU as below.
  __asm MCR P15, 0, temp, c7, c0, 4.

- Exiting condition: (Sleep mode → Normal mode)
  The program will exit from the standby mode when the external wakeup pin is activated or any reset or IRQ/FIQ interrupt is asserted by INTC.

- State definition (In the standby mode):
  - All clocks, PLLs, OSC stop.
  - Main chip power off.
  - DRAM in the self-refresh mode
  - Interrupts are recognized as the wake-up sources.

.

### 4.3.11 Frequency Change Sequence (FCS)

The Frequency Change Sequence (FCS) is used to change the system clock frequency. The system clocks stop during the course of FCS. In this sequence, the frequency is changed from the default condition at initial boot-up. FCS can be used as a power-saving feature that allows GM8126 to operate at the minimum required frequency.

**Entering / Exiting Sequence**: (S → Software action, H → Hardware action)

1. Initiate the interrupt controller (S)
2. Disable the AHB master peripherals to stop the events from coming into the system (S)
3. Program a suitable value to the refresh interval counter of the DDR2 controller (S)
4. Write a new frequency value to PLL1CR[pll1frang, pll1ms, pll1ns, cpuxx_mode] (Offset = 0x30) (S)
5. Set the FCS bit in the PMODE (Offset = 0x0c) register (S)
6. Block the interrupts to CPU (H)
7. Write the CPU power-down command (S)
8. Wait for the idle signal of CPU, CPUOFFN (H)

9. Issue a self-refresh command to the DDR2 SDRAM controller
10. Wait for acknowledge of self-refresh (H)
11. Gate the clocks (pclk, hclk, then fclk) (H)
12. Load the new frequency value to PLL1 (H)
13. Wait for the stable signal of PLL1 (H)
14. Run the clocks free (fclk, then hclk and pclk) (H)
15. Remove the gating of master requests/interrupts (H)
16. Clock PMU for issuing an interrupt to CPU to enter the interrupt subroutine (H)
17. Software must examine the PMSR Register (PMSR[IntFCS]) (Offset: 0x20) to determine the cause for waking up. Software must clear the FCS bit in the PMODE (Offset: 0x0c) register. (S)

- State definition: FCS (Frequency Change Sequence) is implemented to change the PLL1 clock frequency. During the course of FCS, the PLL1 clock stops. This mode is used to change PLL1 from the default condition under the initial boot-up condition. The software must complete the following steps before initiating FCS:
  - Configure the memory controller to ensure that the content of DDR2 is maintained during FCS. The refresh timer of the memory controller must be programmed to match the maximum refresh time associated with the slower of the two frequencies (Current frequency and desired frequency).
  - Disable the LCD controller or configure it to avoid the effects of an interruption in the LCD clocks and data from the application processor.
  - The interrupts generated during FCS will be ignored.
  - In FCS, the PLL1 clock generator is locking to the correct frequency and cannot be used.

## 4.4 Programming Model

### 4.4.1 Summary of Clock and Power Manager Registers

Table 4-3 summarizes the PMU registers.

**Table 4-3. Summary of PMU Registers**

| Address (Offset) | Name | Reset Value |
|---|---|---|
| +0x00 | ID number | 0x8126 _2100 |
| +0x04 | Jump setting | Depend on the jumper setting |
| +0x08 | OSCCR | 0x00b5_8280 |
| +0x0C | PMR | 0x0000_0000 |
| +0x10 | PMCR | 0x0001_0000 |
| +0x20 | PMSR | 0x0000_0000 |
| +0x28 | CSCR | 0x00a0_1000 |
| +0x30 | PLL1CR | Depend on the jumper setting |
| +0x34 | PLL2CR | 0x0000_31b6 |
| +0x38 | AHBMCLKOFF | 0xffff_faf8 |
| +0x3C | APBMCLKOFF | 0xffff_ffff |
| +0x40 | DCSRC0 | 0x0001_1811 |
| +0x44 | DCSRC1 | 0x0000_0000 |
| +0x50 | MFPS0 | 0x0000_0000 |
| +0x54 | MFPS1 | 0x0000_0000 |
| +0x58 | MFPS2 | 0x0000_0000 |
| +0x5C | MFPS3 | 0x0000_0000 |
| +0x60 | MFPS4 | 0x0000_0000 |
| +0x64 | MFPS5 | 0x0000_0000 |
| +0x6C | SCR | 0x003c_c017 |
| +0x70 | Divider setting 0 | 0x1310_4140 |
| +0x74 | Divider setting 1 | 0x0733_4333 |
| +0x78 | Divider setting 2 | 0x2c3d_5555 |
| +0x7C | TVE DAC control/divider setting 3 | 0x9c63_3008 |
| +0x80 | DDR2 PHY control | 0x0000_0007 |
| +0x84 | MAC PHY control | 0x0012_0101 |

| Address (Offset) | Name | Reset Value |
|---|---|---|
| +0x88 | MAC PHY identifier | 0x0000_0000 |
| +0x8C | Audio codec interface control | 0x0000_0000 |
| +0x90 | Audio codec ADC control 0 | 0x0180_0000 |
| +0x94 | Audio codec ADC control 1 | 0x0000_0000 |
| +0x98 | Audio codec DAC control 0 | 0x0000_5000 |
| +0x9C | Audio codec DAC control 1 | 0x0c00_0000 |
| +0xA0 | OTG control resistor | 0x0008_0009 |

### 4.4.2 Register Descriptions

The following sections describe the PMU registers in more details.

The abbreviations below represent the access types used throughout the register descriptions.

- R/W: Read/Write
- RO: Read Only
- W1C: Write 1 Clear
- HR: Hardware Reset (RSTN is asserted.)
- WR: Watchdog Reset

#### 4.4.2.1 ID Number Register (Offset = 0x00)

**Table 4-4. ID Number Register (IDNMBR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:16] | PN | RO | Product name | 0x8126 | - |
| [15:8] | VID | RO | Version ID number | 0x21 | - |
| [7:0] | Reserved | - | - | - | - |

### 4.4.2.2 Jump Setting (Offset = 0x04)

Software can read the jump setting status via this register.

**Table 4-5. Jump Setting**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:8] | - | - | Reserved | - | - |
| [7:6] | jpset[7:6] | RO | {EDP_LINKLED, EDP_SPDLED}<br>00: Reserved<br>10: Reserved<br>01: Firmware Update mode<br>11: Normal mode | Depend on the pin setting | - |
| 5 | jpset[5] | RO | NAND_DATA[7]<br>0: SPI Flash boot<br>1: NAND Flash boot | Depend on the pin setting | - |
| 4 | jpset[4] | RO | UART2_SOUT<br>Internal pull-up for the normal mode | Depend on the pin setting | - |
| 3 | jpset[3] | RO | NAND_DATA[5]<br>Internal pull-up for the normal mode | Depend on the pin setting | - |
| 2 | jpset[2] | RO | UART1_SOUT<br>Internal pull-up for the normal mode | Depend on the pin setting | - |
| 1 | jpset[1] | RO | SPI_FS1<br>pll32k_en<br>0: Disable pll32k<br>1: Enable pll32k | Depend on the pin setting | - |
| 0 | jpet[0] | RO | SPI_FS0<br>PLL1 setting<br>0: PLL1 output frequency is 800 MHz.<br>1: PLL1 output frequency is 400 MHz. | Depending the pin setting | - |

### 4.4.2.3 OSC Control Register (Offset = 0x08)

**Table 4-6. OSC Control Register (OSCCR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:25] | - | - | Reserved | - | - |
| [24:23] | Pll0_icp2 | R/W | Adjustable loop bandwidth for PLL32k | 0x1 | HR/WR |
| [22:20] | Pll0_icp1 | R/W | Adjustable loop bandwidth for PLL32k | 0x3 | HR/WR |
| [19:11] | - | - | Reserved | - | - |
| [10:9] | osch range | R/W | Range select of OSCHIO<br>00: 1 MHz ~ 12 MHz<br>01: 12 MHz ~ 24 MHz<br>10: 24 MHz ~ 42 MHz<br>11: 42 MHz ~ 66 MHz | 0x2 | HR/WR |
| [8:0] | - | - | Reserved | - | - |

### 4.4.2.4 Power Mode Register (Offset = 0x0C)

**Table 4-7. Power Mode Register (PMR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:21] | - | - | Reserved | - | - |
| 20 | ispupintmask | R/W | Mask for the ISP power-up interrupt | 0x0 | HR |
| [19:13] | - | - | Reserved | - | - |
| 12 | isppwroff | R/W | ISP power switch on/off<br>0: ISP power on<br>1: ISP power off | 0 | HR/WR |
| [11:6] | - | - | Reserved | - | - |
| 5 | cpuclkoffdis | R/W | Disable the CPU clock off when entering the Idle mode | 0x0 | HR/WR |
| 4 | oschoffdis | R/W | Disable the OSCH off when entering the standby mode | 0x0 | HR/WR |
| 3 | - | - | Reserved | - | - |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 2 | fcs | R/W | Frequency Change Sequence<br>1: Enter the Frequency Change Sequence<br>When changing the frequency, users must write the CPU coprocessor instruction (Power-down command) after the power-mode instruction.<br>0: Do not enter the Frequency Change Sequence | 0x0 | HR/WR |
| [1:0] | sleep_mode | R/W | Enter the low-power mode when CPU executes the WFI (Wait for Interrupt) instruction<br>00: No effect<br>01: Idle mode (CPU clock off)<br>10: Standby mode (All clocks off, PLL disabled, DRAM in self-refresh mode)<br>11: Sleep mode (Main chip power off, SDRAM in the self-refresh mode) | 0x0 | HR/WR |

### 4.4.2.5 Power Manager Control Register (Offset = 0x10)

**Table 4-8. Power Manager Control Register (PMCR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:17] | - | - | Reserved | - | - |
| 16 | wertc | R/W | RTC alarm control<br>0: Disable the RTC alarm wakeup<br>1: Enable the RTC alarm wakeup | 0x0 | HR/WR |
| [15:0] | - | - | Reserved | - | - |

### 4.4.2.6 Power Manager Status Register (Offset = 0x20)

**Table 4-9. Power Manager Status Register (PMSR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:21] | - | - | Reserved | - | - |
| 20 | Intispup | R/W1C | ISP power up interrupt indication<br>0: NO ISP power up<br>1: ISP power up<br>Write '1' to clear this bit | 0x0 | HR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 19 | Intstdby | R/W1C | Standby mode interrupt indication<br>Write '1' to clear this bit<br>0: No standby interrupt<br>1: Standby interrupt | 0x0 | HR |
| 18 | Intidle | R/W1C | Idle mode interrupt indication<br>Write '1' to clear this bit<br>0: No idle interrupt<br>1: Idle interrupt | 0x0 | HR |
| 17 | Intfcs | R/W1C | FCS mode interrupt indication<br>Write '1' to clear this bit<br>0: No FCS interrupt<br>1: FCS interrupt | 0x0 | HR |
| [16:11] | - | - | Reserved | - | - |
| 10 | smr | R/W1C | Wake up from the sleep mode<br>1: Sleep mode has occurred since the last time the CPU or hardware reset cleared this bit.<br>0: Sleep mode has not occurred since the last time the CPU or hardware reset cleared this bit.<br>Cleared by hardware reset and by setting to '1'. | 0x0 | HR |
| 9 | wdr | R/W1C | Reboot by the watchdog reset<br>0: Watchdog reset has not occurred since the last CPU or since the hardware reset cleared this bit.<br>1: Watchdog reset has occurred since the last CPU or since the hardware cleared this bit.<br>Set this bit when the watchdog reset occurs<br>Cleared by software programming | 0x0 | HR |
| [8:0] | - | - | Reserved | - | - |

### 4.4.2.7 Clock Source Control Register (Offset = 0x28)

**Table 4-10. Clock Source Control Register (CSCR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 31 | extclk3_sel | R/W | External clock output divider source 3<br>0: Select the PLL2 output<br>1: Select the 30-MHz clock | 0x0 | HR/WR |
| 30 | extclk2_sel | R/W | External clock output divider source 2<br>0: Select the PLL2 output<br>1: Select the 30-MHz clock | 0x0 | HR/WR |
| 29 | extclk1_sel | R/W | External clock output divider source 1<br>0: Select the PLL2 output<br>1: Select the 30-MHz clock | 0x0 | HR/WR |
| 28 | extclk0_sel | R/W | External clock output divider source 0<br>0: Select the PLL2 output<br>1: Select the 30-MHz clock | 0x0 | HR/WR |
| [27:24] | - | - | Reserved | - | - |
| [23:22] | vcap1clk_sel | R/W | The bits select VCAP1 pclk as the input source.<br>0: Select CAP1_CLK<br>1: Select inverse clk of CAP1_CLK<br>2: Select the PLL1 divider output<br>3: Reserved | 0x2 | HR/WR |
| [21:20] | vcap0clk_sel | R/W | These bits select VCAP0 pclk as the input source.<br>0: Select CAP0_CLK<br>1: Select the inverse clock, CAP0_CLK<br>2: Select PLL1 divider output<br>3: Reserved | 0x2 | HR/WR |
| [19:18] | mcp210clk_sel | R/W | These bits select the mcp210 frequency divider clock source.<br>0: Select the PLL1 output<br>1: Select HCLK<br>2: Select PLL1/2.5<br>3: Reserved | 0x0 | HR/WR |
| [17:16] | - | - | Reserved | - | - |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 15 | uarclk_sel | - | The bit selects UART as the clock source.<br>0: Select the 30-MHz clock<br>1: Select the PLL2 divider output | 0x0 | HR/WR |
| [14:12] | - | - | Reserved | - | - |
| 11 | cmosclk_sel | R/W | This bit selects the CMOS/CAPTURE frequency divider clock source.<br>0: Select the PLL2 output<br>1: Select the 30-MHz OSC output | 0x0 | HR/WR |
| 10 | adcclk_sel | R/W | The bit selects the SAR-ADC frequency divider clock source.<br>0: Select the PLL2 output<br>1: Select the 30-MHz clock | 0x0 | HR/WR |
| 9 | - | - | Reserved | - | - |
| 8 | macphyclk_sel | R/W | This bit selects the MAC frequency divider clock source.<br>0: Select the PLL1 output<br>1: Select HCLK | 0x0 | HR/WR |
| 7 | mcp100clk_sel | R/W | This bit selects the mcp100 frequency divider clock source.<br>0: Select the PLL1 output<br>1: Select HCLK | 0x0 | HR/WR |
| 6 | - | - | Reserved | - | - |
| [5:1] | - | - | Reserved | - | - |
| 0 | debug_en | R/W | 0: Disable the debug mode<br>1: Enable the debug mode<br>Note: The software will not change the default value in the normal operation. | 0x0 | HR/WR |

### 4.4.2.8 PLL1 Control Register (Offset = 0x30)

**Table 4-11. PLL1 Control Register (PLL1CR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31] | cpuxx_mode[10] | WO | DDR2 DLL frequency range<br>jpset[0] = 1'b1, 350 MHz ~ 600 MHz<br>jpset[0] = 1'b0, 600 MHz ~ 810 MHz | Depending on the jumper setting | WO |
| [30:22] | cpuxx_mode[9:0] | R/W | jpset[0] = 1'b1 → 9'b000_0101_00<br>FCLK = PLL1 output/2 (200 MHz)<br>HCLK = PLL1 output/2 (200 MHz)<br>DDRCLK2X = PLL1 output (400 MHz)<br>jpset[0] = 1'b0 → 9'b110_1010_00<br>FCLK = PLL1 output/1.5 (533 MHz)<br>HCLK = PLL1 output/3 (266 MHz)<br>DDRCLK2X = PLL1 output (800 MHz) | Depending on the jumper setting | HR/WR |
| [21:20] | pll1frange[1:0] | R/W | PLL1 output frequency range<br>00: 50 MHz ~ 125 MHz<br>01: 125 MHz ~ 250 MHz<br>10: 250 MHz ~ 500 MHz<br>11: 500 MHz ~ 1000 MHz | Depending on the jumper setting | HR/WR |
| [19:17] | - | - | Reserved | - | - |
| [16:12] | pll1ms[4:0] | R/W | This field indicates the M value of the embedded PLL1 to control the frequency output of PLL1.<br>Output frequency of PLL1 = 30 MHz * pll1 ns/pll1 ms | Depending on the jump setting | HR/WR |
| [11:3] | pll1ns[8:0] | R/W | This field indicates the N value of the embedded PLL1 to control the frequency output of PLL1. | Depending on the jump setting | HR/WR |
| 2 | - | - | Reserved | - | - |
| 1 | pll1stable | R/W | This bit indicates whether PLL1 is stable or not. | 0x0 | HR/WR |
| 0 | pll1en | R/W | 0: Enable PLL1<br>1: Disable PLL1 | 0x0 | R/W |

### 4.4.2.9 PLL2 Control Register (Offset = 0x34)

**Table 4-12. PLL2 Control Register (PLL2CR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:17] | | - | Reserved | - | - |
| [16:12] | pll2ms[4:0] | R/W | This field indicates the M value of the embedded PLL2 to control the frequency output of PLL2.<br>Output frequency of PLL2 = 30 MHz * pll2 ns/pll2 ms | 0x3 | HR/WR |
| [11:3] | pll2ns[8:0] | R/W | This field indicates the N value of the embedded PLL2 to control the frequency output of PLL2<br>The default output frequency is 540 MHz. | 0x36 | HR/WR |
| [2:1] | pll2frange[1:0] | R/W | PLL1 output frequency range<br>00: 50 MHz ~ 125 MHz<br>01: 125 MHz ~ 250 MHz<br>10: 250 MHz ~ 500 MHz<br>11: 500 MHz ~ 1000 MHz | 0x3 | HR/WR |
| 0 | pll2en | R/W | 0: Enable PLL2<br>1: Disable PLL2<br>Note: The locking time of PLL is 500 µs. | 0x0 | HR/WR |

### 4.4.2.10 AHB Module Clock Off Control Register (Offset = 0x38)

**Table 4-13. AHB Module Clock Off Control Register (AHBMCLKOFF)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:27] | - | - | Reserved | - | - |
| 26 | cap1inoff | R/W | 0: CAPTURE1 pixel clock on<br>1: CAPTURE1 pixel clock off | 0x1 | HR/WR |
| 25 | cap0inoff | R/W | 0: CAPTURE0 pixel clock on<br>1: CAPTURE0 pixel clock off | 0x1 | HR/WR |
| 24 | cap1off | R/W | 0: CAPTURE1 HCLK clock on<br>1: CAPTURE1 HCLK clock off | 0x1 | HR/WR |
| 23 | cap0off | R/W | 0: CAPTURE0 HCLK clock on<br>1: CAPTURE0 HCLK clock off | 0x1 | HR/WR |
| 22 | bmeoff | R/W | 0: BME HCLK clock on<br>1: BME HCLK clock off | 0x1 | HR/WR |
| 21 | ispoff | R/W | 0: ISP HCLK clock on<br>1: ISP HCLK clock off | 0x1 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 20 | di3doff | R/W | 0: DI3D clock on<br>1: DI3D clock off | 0x1 | HR/WR |
| [19:17] | - | - | Reserved | - | - |
| 16 | fotgoff | R/W | 0: OTG clock on<br>1: OTG clock off | 0x1 | HR/WR |
| 15 | - | - | Reserved | - | - |
| 14 | scaroff | R/W | 0: SCAR clock on<br>1: SCAR clock off | 0x1 | HR/WR |
| 13 | ct656off | R/W | 0: CT656 clock on<br>1: CT656 clock off | 0x1 | HR/WR |
| 12 | sdcoff | R/W | 0: SDC clock on<br>1: SDC clock off | 0x1 | HR/WR |
| 11 | dmaoff | R/W | 0: DMA clock on<br>1: DMA clock off | 0x1 | HR/WR |
| 10 | macphyoff | R/W | 0: MAC PHY clock on<br>1: MAC PHY clock off | 0x0 | HR/WR |
| 9 | aesoff | R/W | 0: AES clock on<br>1: AES clock off | 0x1 | HR/WR |
| 8 | macoff | R/W | 0: MAC clock on<br>1: MAC clock off | 0x0 | HR/WR |
| 7 | lcdoff | R/W | 0: LCD clock on<br>1: LCD clock off | 0x1 | HR/WR |
| 6 | mcp100off | R/W | 0: IDE clock on<br>1: IDE clock off | 0x1 | HR/WR |
| 5 | nandoff | R/W | 0: NAND clock on<br>1: NAND clock off | 0x1 | HR/WR |
| 4 | mcp210off | R/W | 0: MCP210 clock on<br>1: MCP210 clock off | 0x1 | HR/WR |
| 3 | ddr2off | R/W | 0: DDR2 clock on<br>1: DDR2 clock off | 0x1 | HR/WR |
| 2 | memcoff | R/W | 0: Memory clock on<br>1: Memory clock off | 0x0 | HR/WR |
| [1:0] | - | - | Reserved | - | - |

### 4.4.2.11 APB Module Clock Off Control Register (Offset = 0x3C)

**Table 4-14. APB Module Clock Off Control Register (APBMCLKOFF)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [30:28] | - | - | Reserved | - | - |
| 27 | gpcnoff | R/W | 0: GPCN clock on<br>1: GPCN clock off | 0x1 | HR/WR |
| 26 | cmosoff | R/W | 0: CMOS clock on<br>1: CMOS clock off | 0x1 | HR/WR |
| 25 | adcoff | R/W | 0: ADC clock on<br>1: ADC clock off | 0x1 | HR/WR |
| 24 | pwmoff | R/W | 0: PWM clock on<br>1: PWM codec clock off | 0x1 | HR/WR |
| 23 | audiooff | R/W | 0: AUDIO codec clock on<br>1: AUDIO codec clock off | 0x1 | HR/WR |
| 22 | extclk3off | R/W | 0: EXT3 clock on<br>1: EXT3 clock off | 0x1 | HR/WR |
| 21 | extclk2off | R/W | 0: EXT2 clock on<br>1: EXT2 clock off | 0x1 | HR/WR |
| 20 | extclk1off | R/W | 0: EXT1 clock on<br>1: EXT1 clock off | 0x1 | HR/WR |
| 19 | extclk0off | R/W | 0: EXT0 clock on<br>1: EXT0 clock off | 0x1 | HR/WR |
| 18 | irdetoff | R/W | 0: IRDet clock on<br>1: IRDet clock off | 0x1 | HR/WR |
| 17 | PWMoff | R/W | 0: PWM clock on<br>1: PWM clock off | 0x1 | HR/WR |
| 16 | RTCoff | R/W | 0: RTC clock on<br>1: RTC clock off | 0x1 | HR/WR |
| 15 | wdtoff | R/W | 0: WDT clock on<br>1: WDT clock off | 0x1 | HR/WR |
| 14 | tmroff | R/W | 0: TMR clock on<br>1: TMR clock off | 0x1 | HR/WR |
| 13 | i2coff | R/W | 0: $I^2C$ clock on<br>1: $I^2C$ clock off | 0x1 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 12 | intcoff | R/W | 0: INTC clock on<br>1: INTC clock off | 0x1 | HR/WR |
| 11 | gpio2off | R/W | 0: GPIO2 clock on<br>1: GPIO2 clock off | 0x1 | HR/WR |
| 10 | gpio1off | R/W | 0: GPIO1 clock on<br>1: GPIO1 clock off | 0x1 | HR/WR |
| 9 | gpio0off | R/W | 0: GPIO0 clock on<br>1: GPIO0 clock off | 0x1 | HR/WR |
| 8 | sdcmoff | R/W | 0: SDC pclk on<br>1: SDC pclk off | 0x1 | HR/WR |
| 7 | ssp2off | R/W | 0: SSP2 clock on<br>1: SSP2 clock off | 0x1 | HR/WR |
| 6 | ssp1off | R/W | 0: SSP1 clock on<br>1: SSP1 clock off | 0x1 | HR/WR |
| 5 | ssp0off | R/W | 0: SSP0 clock on<br>1: SSP0 clock off | 0x1 | HR/WR |
| 4 | uart4off | R/W | 0: UART4 clock on<br>1: UART4 clock off | 0x1 | HR/WR |
| 3 | uart3off | R/W | 0: UART3 clock on<br>1: UART3 clock off | 0x1 | HR/WR |
| 2 | uart2off | R/W | 0: UART2 clock on<br>1: UART2 clock off | 0x1 | HR/WR |
| 1 | uart1off | R/W | 0: UART1 clock on<br>1: UART1 clock off | 0x1 | HR/WR |
| 0 | uart0off | R/W | 0: UART0 clock on<br>1: UART0 clock off | 0x1 | HR/WR |

### 4.4.2.12 Driving Capability and Slew Rate Control Register 0 (Offset = 0x40)

**Table 4-15. Driving Capability and Slew Rate Control Register 0 (DCSRCR0)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | - | - | Reserved | - | - |
| [29:26] | cap_dcsr | R/W | capclk_dcsr[4:1] sets the output driving capability and output slew rate.<br>The field includes CAP_CLKOUT.<br>capclk_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>capclk_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow<br>capclk_dcsr[4]: Schmitt-trigger control<br>0: Normal<br>1: Schmitt-trigger | 0x0 | HR/WR |
| [25:21] | cap1_dcsr | R/W | cap1_dcsr[4:0] sets the output driving capability and output slew rate.<br>This field includes CAP1_CLK, CAP1_VS, CAP1_HS, CAP1_FIELD, and CAP1_D[7:0].<br>cap0_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>cap1_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>cap1_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow<br>cap1_dcsr[4]: Schmitt-trigger control of CAP1_CLK<br>0: Normal operation<br>1: Schmitt-trigger | 0x1 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [20:16] | cap0_dcsr | R/W | cap0_dcsr[4:0] sets the output driving capability and output slew rate.<br>This field includes CAP0_CLK, CAP0_VS, CAP0_HS, CAP0_FIELD, and CAP0_D[7:0].<br>cap0_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>cap0_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>cap0_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow<br>cap0_dcsr[4]: Schmitt-trigger control of CAP0_CLK<br>0: Normal operation<br>1: Schmitt-trigger | 0x1 | HR/WR |
| [15:12] | mac_dcsr | R/W | mac_dcsr[3:0] sets the MAC output driving capability and output slew rate.<br>This field includes TV_D[15:0], EDP_SPDLED, and EDP_LINKLED.<br>mac_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>mac_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>mac_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x1 | HR/WR |
| [11:8] | - | - | Reserved | - | - |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [7:4] | ice_dcsr | R/W | ice_dcsr[3:0] controls the output driving capability and output slew rate.<br>This field includes NTRST, TCK, TDI, TDO, and TMS.<br>ice_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>ice_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>ice_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x1 | HR/WR |
| [3:0] | nand_dcsr | R/W | nand_dcsr[3:0] controls the the output driving capability and output slew rate.<br>This field includes NAND_REN, NAND_WEN, NAND_CLE, NAND_ALE, NAND_D[7:0], and NAND_CEN[1:0].<br>nand_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>nand_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>nand_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x1 | HR/WR |

### 4.4.2.13 Driving Capability and Slew Rate Control Register 1 (Offset = 0x44)

**Table 4-16. Driving Capability and Slew Rate Control Register 1 (DCSRCR1)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:24] | - | - | Reserved | - | - |
| [23:20] | sdc_dcsr | R/W | sdc_dcsr[3:0] controls the output driving capability and output slew rate.<br>This field includes SD_CD, SD_CLK, and SD_CMD_RSP.<br>sdc_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>sdc_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>sdc_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x0 | HR/WR |
| [19:16] | - | R/W | Reserved | - | - |
| [15:12] | I2c_dcsr | R/W | i2c_dcsr[3:0] controls the output driving capability and output slew rate.<br>This field includes I2C_SCL and I2C_SDA.<br>i2c_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>i2c_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>i2c_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [11:8] | I2s_dcsr | R/W | i2s_dcsr[3:0] controls the output driving capability and output slew rate.<br>This field includes SPI_FS0, SPI_FS1, SPI_SCLK, SPI_TXD, SPI_RXD, I2S_FS, I2S_SCLK, I2S_TXD, and I2S_RXD.<br>i2s_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>i2s_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>i2s_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x0 | HR/WR |
| [7:4] | pwm_dcsr | R/W | pwm_dcsr[3:0] controls the output driving capability and output slew rate.<br>This field includes PWM[0], PWM[1], PWM[2], and PWM[3].<br>pwm_dcsr[0]: Schmitt-trigger control<br>0: Normal operation<br>1: Schmitt-trigger<br>pwm_dcsr[2:1]: Driving capability control<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>pwm_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [3:0] | uart_dcsr | R/W | The uart_dcsr[3:0] signal sets the output driving capability and output slew rate.<br>This field includes UART0_SOUT, UART0_SIN, UART1_SOUT, UART1_SIN, UART2_SOUT, UART2_SIN, UART3_SOUT, UART3_SIN, UART4_SOUT, and UART4_SIN.<br>uart_dcsr[0]:<br>0: Normal operation<br>1: Schmitt-trigger<br>uart_dcsr[2:1]:<br>00: 4 mA<br>01: 8 mA<br>10: 12 mA<br>11: 16 mA<br>uart_dcsr[3]: Slew rate control<br>0: Fast<br>1: Slow | 0x0 | HR/WR |

### 4.4.2.14 Multi-function Port Setting Register 0 (Offset = 0x50)

**Table 4-17. Multi-function Port Setting Register 0 (MFPSR0)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | SPI_gpio_mux_txd | R/W | Mux control of the SPI_TXD and GPIO_0[18] pins<br>00: GPIO_0[18] pin is mux-ed out.<br>01: SPI_TXD pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [29:28] | SPI_gpio_mux_clk | R/W | Mux control of the SPI_SCLK and GPIO_0[17] pins<br>00: GPIO_0[17] pin is mux-ed out.<br>01: SPI_SCLK pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [27:26] | SPI_gpio_mux_rxd | R/W | Mux control of the SPI_RXD and GPIO_0[16] pins<br>00: GPIO_0[16] pin is mux-ed out.<br>01: SPI_RXD pin is mux-ed in.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [25:24] | uart4_gpio_mux_sout | R/W | Mux control of the UART4_SOUT and GPIO_0[13] pins<br>00: GPIO_0[13] pin is mux-ed out.<br>01: UART4_SOUT pin is mux-ed out.<br>10: UART1_DCDN is mux-ed in.<br>11: DMIC_CLK is mux-ed in. | 0x0 | HR/WR |
| [23:22] | Uart4_gpio_mux_sin | R/W | Mux control of the UART4_SIN and GPIO_0[12] pins<br>00: GPIO_0[12] pin is mux-ed out.<br>01: UART4_SIN pin is mux-ed in.<br>10: UART1_RIN is mux-ed in.<br>11: DMIC_DATA is mux-ed in. | 0x0 | HR/WR |
| [21:20] | uart3_gpio_mux_sout | R/W | Mux control of the UART3_SOUT and GPIO_0[11] pins<br>00: GPIO_0[11] pin is mux-ed out.<br>01: UART3_SOUT pin is mux-ed out.<br>10: UART1_DTRN is mux-ed out.<br>11: UART1_DTRN is mux-ed out (RS-485). | 0x0 | HR/WR |
| [19:18] | uart3_gpio_mux_sin | R/W | Mux control of the UART3_SIN and GPIO_0[10] pins<br>00: GPIO_0[10] pin is mux-ed out.<br>01: UART3_SIN pin is mux-ed in.<br>10: UART1_DSRN is mux-ed in.<br>11: UART4_DTRN is mux-ed out. | 0x2 | HR/WR |
| [17:16] | uart2_gpio_mux_sout | R/W | Mux control of the UART2_SOUT and GPIO_0[9] pins<br>00: GPIO_0[9] pin is mux-ed out.<br>01: UART2_SOUT pin is mux-ed out.<br>10: UART1_RTSN is mux-ed out.<br>11: UART1_DTRN is mux-ed out. | 0x2 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [15:14] | uart2_gpio_mux_sin | R/W | Mux control of the UAR2_SIN and GPIO_0[8] pins<br>00: GPIO_0[8] pin is mux-ed out.<br>01: UART2_SIN pin is mux-ed in.<br>10: UART1_CTSN is mux-ed in.<br>11: Reserved | 0x2 | HR/WR |
| [13:12] | uart1_gpio_mux_sout | R/W | Mux control of the UART1_SOUT and GPIO_0[7] pins<br>00: GPIO_0[7] pin is mux-ed out.<br>01: UART1_SOUT pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x1 | HR/WR |
| [11:10] | uart1_gpio_mux_sin | R/W | Mux control of the UART1_SIN and GPIO_0[6] pins<br>00: GPIO_0[6] pin is mux-ed out.<br>01: UART1_SIN pin is mux-ed in.<br>10: Reserved<br>11: Reserved | 0x1 | HR/WR |
| 9 | Uart0_gpio_mux_sout | R/W | Mux control of the UART0_SOUT and GPIO_0[5] pins<br>0: UART0_SOUT pin is mux-ed out.<br>1: GPIO_0[5] pin is mux-ed out. | 0x1 | HR/WR |
| 8 | Uart0_gpio_mux_sin | R/W | Mux control of the UART0_SIN and GPIO_0[4] pins<br>0: UART0_SIN pin is mux-ed in.<br>1: GPIO_0[4] pin is mux-ed out. | 0x0 | HR/WR |
| [7:6] | pwm_pio_mux3 | R/W | Mux control of the PWM[3], ext_clk[3], and GPIO_0[3] pins<br>00: GPIO_0[3] pin is mux-ed out.<br>01: PWM[3] pin is mux-ed out.<br>10: ext_clk[3] is mux-ed out.<br>11: IR-Det[11]/GPCN[11] is mux-ed in. | 0x0 | HR/WR |
| [5:4] | pwm_pio_mux0 | R/W | Mux control of the PWM[2], ext_clk[2], and GPIO_0[2] pins<br>00: GPIO_0[2] pin is mux-ed out.<br>01: PWM[2] pin is mux-ed out.<br>10: ext_clk[2] is mux-ed out.<br>11: IR-Det[10]/GPCN[10] is mux-ed in. | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [3:2] | pwm_pio_mux1 | R/W | Mux control of the PWM[1], ext_clk[1], and GPIO_0[1] pins<br>00: GPIO_0[1] pin is mux-ed out.<br>01: PWM[1] pin is mux-ed out.<br>10: ext_clk[1] is mux-ed out.<br>11: DMIC_CLK is mux-ed in. | 0x0 | HR/WR |
| [1:0] | pwm_pio_mux0 | R/W | Mux control of the PWM[0], ext_clk[0], and GPIO_0[0] pins<br>00: GPIO_0[0] pin is mux-ed out.<br>01: PWM[0] pin is mux-ed out.<br>10: ext_clk[0] is mux-ed out.<br>11: DMIC_DATAis mux-ed in. | 0x0 | HR/WR |

### 4.4.2.15 Multi-function Port Setting Register 1 (Offset = 0x54)

**Table 4-18. Multi-function Port Setting Register 1 (MFPSR1)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | cap0_gpio_mux_d7 | R/W | Mux control of the CAP0_D[7] and GPIO_1[11] pins<br>00: GPIO_1[11] pin is mux-ed out.<br>01: CAP0_D[7] pin is mux-ed in.<br>10: Bayer data[11] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [29:28] | cap0_gpio_mux_d6 | R/W | Mux control of the CAP0_D[6] and GPIO_1[10] pins<br>00: GPIO_1[10] pin is mux-ed out.<br>01: CAP0_D[6] pin is mux-ed in.<br>10: Bayer data[10] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [27:26] | cap0_gpio_mux_d5 | R/W | Mux control of the CAP0_D[5] and GPIO_1[9] pins<br>00: GPIO_1[9] pin is mux-ed out.<br>01: CAP0_D[5] pin is mux-ed in.<br>10: Bayer data[9] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [25:24] | cap0_gpio_mux_d4 | R/W | Mux control of the CAP0_D[4] and GPIO_1[8] pins<br>00: GPIO_1[8] pin is mux-ed out.<br>01: CAP0_D[4] pin is mux-ed in.<br>10: Bayer data[8] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [23:22] | cap0_gpio_mux_d3 | R/W | Mux control of the CAP0_D[3] and GPIO_1[7] pins<br>00: GPIO_1[7] pin is mux-ed out.<br>01: CAP0_D[3] pin is mux-ed in.<br>10: Bayer data[7] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [21:20] | cap0_gpio_mux_d2 | R/W | Mux control of the CAP0_D[2] and GPIO_1[6] pins<br>00: GPIO_1[6] pin is mux-ed out.<br>01: CAP0_D[2] pin is mux-ed in.<br>10: Bayer data[6] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [19:18] | cap0_gpio_mux_d1 | R/W | Mux control of the CAP0_D[1] and GPIO_1[5] pins<br>00: GPIO_1[5] pin is mux-ed out.<br>01: CAP0_D[1] pin is mux-ed in.<br>10: Bayer data[5] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [17:16] | cap0_gpio_mux_d0 | R/W | Mux control of the CAP0_D[0] and GPIO_1[4] pins<br>00: GPIO_1[4] pin is mux-ed out.<br>01: CAP0_D[0] pin is mux-ed out.<br>10: Bayer data[4] pin is mux-ed in.<br>11: Reserved. | 0x0 | HR/WR |
| [15:14] | cap0_gpio_mux_clk | R/W | Mux control of the CAP0_CLK and GPIO_1[3] pins<br>00: GPIO_1[3] pin is mux-ed out.<br>01: CAP0_CLK pin is mux-ed in.<br>10: Bayer clk pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [13:12] | cap0_gpio_mux_vs | R/W | Mux control of the CAP0_VS and GPIO_1[2] pins<br>00: GPIO_1[2] pin is mux-ed out.<br>01: CAP0_VS pin is mux-ed in.<br>10: Bayer frame pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [11:10] | cap0_gpio_mux_hs | R/W | Mux control of the CAP0_HS and GPIO_1[1] pins<br>00: GPIO_1[1] pin is mux-ed out.<br>01: CAP0_HS pin is mux-ed in.<br>10: Bayer line pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [9:8] | cap0_gpio_mux_fld | R/W | Mux control of th CAP0_FIELD and O_1[0] pins<br>00: GPIO_1[0] pin is mux-ed out.<br>01: CAP0_FIELD pin is mux-ed in.<br>10: Bayer strobe pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [7:6] | spi_gpio_mux_fs1 | R/W | Mux control of the SPI_FS1 and GPIO_0[15] pins<br>00: GPIO_0[15] pin is mux-ed out.<br>01: SPI_FS1 pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [5:4] | spi_gpio_mux_fs0 | R/W | Mux control of the SPI_FS0 and GPIO_0[14] pins<br>00: GPIO_0[14] pin is mux-ed out.<br>01: SPI_FS0 pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| 3 | - | - | Reserved | - | - |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [2:0] | spi_fs_dmux | R/W | De-mux the SPI frame sync.<br>000: Select ssp0_fs0<br>001: Select ssp0_fs1<br>010: Select ssp0_fs2<br>011: Select ssp0_fs3<br>100: Select ssp0_fs4<br>101: Select ssp0_fs5<br>110: Reserved<br>111: Reserved | 0x0 | HR/WR |

### 4.4.2.16 Multi-function Port Setting Register 2 (Offset = 0x58)

**Table 4-19. Multi-function Port Setting Register 2 (MFPSR2)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 31 | tv_gpio_mux_ie | R/W | TV GPIO IE selection<br>0: Controlled by system<br>1: Always enable | 0x0 | HR/WR |
| 30 | ice_gpio_mux_1 | R/W | Mux control of the ICE EDBGRQ and GPIO_2[12] pins<br>0: EDBGRQ pin is mux-ed in.<br>1: GPIO_2[12] pin is mux-ed out. | 0x0 | HR/WR |
| 29 | ice_gpio_mux_0 | R/W | Mux control of the ICE DBGACK and GPIO_2[11] pins<br>0: DBGACK pin is mux-ed out.<br>1: GPIO_2[11] pin is mux-ed out. | 0x0 | HR/WR |
| 28 | sd_gpio_mux_cd | R/W | Mux control of the SD_CD and GPIO_1[27] pins<br>0: GPIO_1[27] pin is mux-ed out.<br>1: SD_CD pin is mux-ed out. | 0x0 | HR/WR |
| 27 | sd_gpio_mux_clk | R/W | Mux control of the SD_CLK and GPIO_1[26] pins<br>0: GPIO_1[26] pin is mux-ed out.<br>1: SD_CLK pin is mux-ed out. | 0x0 | HR/WR |
| 26 | sd_gpio_mux_cmd | R/W | Mux control of the SD_CMD_RSP and GPIO_1[25] pins<br>0: GPIO_1[25] pin is mux-ed out.<br>1: SD_CMD_RSP pin is mux-ed out. | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 25 | i2c_gpio_mux_1 | R/W | Mux control of the I2C_SCL and GPIO_0[24] pins<br>0: I2C_SCL pin is mux-ed out.<br>1: GPIO_0[24] pin is mux-ed out. | 0x0 | HR/WR |
| 24 | i2c_gpio_mux_0 | R/W | Mux control of the I2C_SDA and GPIO_0[23] pins<br>0: I2C_SDA pin is mux-ed out.<br>1: GPIO_0[23] pin is mux-ed out. | 0x0 | HR/WR |
| [23:22] | cap1_gpio_mux_d7 | R/W | Mux control of the CAP1_D[7] and GPIO_1[21] pins<br>00: GPIO_1[21] pin is mux-ed out.<br>01: CAP1_D[7] pin is mux-ed in.<br>10: Bayer data[3] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [21:20] | cap1_gpio_mux_d6 | R/W | Mux control of the CAP1_D[6] and GPIO_1[20] pins<br>00: GPIO_1[20] pin is mux-ed out.<br>01: CAP1_D[6] pin is mux-ed in.<br>10: Bayer data[2] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [19:18] | cap1_gpio_mux_d5 | R/W | Mux control of the CAP1_D[5] and GPIO_1[19] pins<br>00: GPIO_1[19] pin is mux-ed out.<br>01: CAP1_D[5] pin is mux-ed in.<br>10: Bayer data[1] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [17:16] | cap1_gpio_mux_d4 | R/W | Mux control of the CAP1_D[4] and GPIO_1[18] pins<br>00: GPIO_1[18] pin is mux-ed out.<br>01: CAP1_D[4] pin is mux-ed in.<br>10: Bayer data[0] pin is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [15:13] | cap1_gpio_mux_d3 | R/W | Mux control of the CAP1_D[3] and GPIO_1[17] pins<br>000: GPIO_1[17] pin is mux-ed out.<br>001: CAP1_D[3] pin is mux-ed in.<br>010: ct656[3] is mux-ed out.<br>011: .Reserved<br>100: PWM[3] is mux-ed out.<br>101: IR-Det[3]/GPCN[3] is mux-ed in.<br>110: SPI_FS6<br>111: I2C_SCL | 0x0 | HR/WR |
| [12:10] | cap1_gpio_mux_d2 | R/W | Mux control of the CAP1_D[2] and GPIO_1[16] pins<br>000: GPIO_1[16] pin is mux-ed out.<br>001: CAP1_D[2] pin is mux-ed in.<br>010: ct656[2] is mux-ed out.<br>011: Reserved<br>100: PWM[2] is mux-ed out.<br>101: IR-Det[2]/GPCN[2] is mux-ed in.<br>110: SPI_RXD<br>111: I2C_SDA | 0x0 | HR/WR |
| [9:7] | cap1_gpio_mux_d1 | R/W | Mux control of the CAP1_D[1] and GPIO_1[15] pins<br>000: GPIO_1[15] pin is mux-ed out.<br>001: CAP1_D[1] pin is mux-ed in.<br>010: ct656[1] is mux-ed out.<br>011: Reserved<br>100: PWM[2] is mux-ed out.<br>101: IR-Det[2]/GPCN[2] is mux-ed in.<br>110: SPI_TXD<br>111: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [6:4] | cap1_gpio_mux_d0 | R/W | Mux control of the CAP1_D[0] and GPIO_1[14] pins<br>000: GPIO_1[14] pin is mux-ed out.<br>001: CAP1_D[0] pin is mux-ed in.<br>010: ct656[0] is mux-ed out.<br>011: Reserved.<br>100: PWM[0] is mux-ed out.<br>101: IR-Det[0]/GPCN[0] is mux-ed in.<br>110: SPI_SCLK<br>111: Reserved | 0x0 | HR/WR |
| [3:2] | cap1_gpio_mux_clk | R/W | Mux control of the CAP1_CLK and GPIO_1[13] pins<br>00: GPIO_1[13] pin is mux-ed out.<br>01: CAP1_CLK pin is mux-ed in.<br>10: ct656 clock is mux-ed out. | 0x0 | HR/WR |
| [1:0] | cap_clkout_mux | R/W | Mux control of the CAP_CLKOUT and GPIO_1[12] pins<br>00: GPIO_1[12] pin is mux-ed out.<br>01: Reserved<br>10: CMOS sensor clock output<br>11: Reserved | 0x0 | HR/WR |

### 4.4.2.17 Multi-function Port Setting Register 3 (Offset = 0x5C)

**Table 4-20. Multi-function Port Setting Register 3 (MFPSR3)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | edp_linkled_gpio_mux | R/W | Mux control of the EDP_LINKLED and GPIO_2[14] pins<br>00: GPIO_2[14] pin is mux-ed out.<br>01: EDP_LINKLED pin is mux-ed out.<br>10: IR-Det[18] is mux-ed in.<br>11: Reserved | 0x0 | HR/WR |
| [29:28] | edp_spdled_gpio_mux | R/W | Mux control of the EDP_SPDLED and GPIO_2[13] pins<br>00: GPIO_2[13] pin is mux-ed out.<br>01: EDP_SPDLED pin is mux-ed out.<br>10: IR-Det[17] is mux-ed in.<br>11: Reserve | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [27:26] | nand_gpio_mux_wen | R/W | Mux control of the NAND_WEN and GPIO_2[5] pins<br>00: GPIO_2[5] pin is mux-ed out.<br>01: NAND_WEN pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [25:24] | nand_gpio_mux_ale | R/W | Mux control of the NAND_ALE and GPIO_2[4] pins<br>00: GPIO_2[4] pin is mux-ed out.<br>01: NAND_ALE pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [23:22] | nand_gpio_mux_cle | R/W | Mux control of the NAND_CLE and GPIO_2[3] pins<br>00: GPIO_2[3] pin is mux-ed out.<br>01: NAND_CLE pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [21:20] | nand_gpio_mux_cen1 | R/W | Mux control of the NAND_CEN[1] and GPIO_2[2] pins<br>00: GPIO_2[2] pin is mux-ed out.<br>01: NAND_CEN1 pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [19:18] | nand_gpio_mux_cen0 | R/W | Mux control of the NAND_CEN0 and GPIO_2[1] pins<br>00: GPIO_2[1] pin is mux-ed out.<br>01: NAND_CEN0 pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [17:16] | nand_gpio_mux_ren | R/W | Mux control of the NAND_REN and GPIO_2[0] pins<br>00: GPIO_2[0] pin is mux-ed out.<br>01: NAND_REN pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [15:14] | nand_gpio_mux_d7 | R/W | Mux control of the NAND_D[7] and GPIO_2[9] pins<br>00: GPIO_2[9] pin is mux-ed out.<br>01: NAND_D[7] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [13:12] | nand_gpio_mux_d6 | R/W | Mux control of the NAND_D[6] and GPIO_2[8] pins<br>00: GPIO_2[8] pin is mux-ed out.<br>01: NAND_D[6] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [11:10] | nand_gpio_mux_d5 | R/W | Mux control of the NAND_D[5] and GPIO_2[7] pins<br>00: GPIO_2[7] pin is mux-ed out.<br>01: NAND_D[5] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [9:8] | nand_gpio_mux_d4 | R/W | Mux control of the NAND_D[4] and GPIO_2[6] pins<br>00: GPIO_2[6] pin is mux-ed out.<br>01: NAND_D[4] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [7:6] | nand_gpio_mux_d3 | R/W | Mux control of the NAND_D[3] and GPIO_1[31] pins<br>00: GPIO_1[31] pin is mux-ed out.<br>01: NAND_D[3] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [5:4] | nand_gpio_mux_d2 | R/W | Mux control of the NAND_D[2] and GPIO_1[30] pins<br>00: GPIO_1[30] pin is mux-ed out.<br>01: NAND_D[2] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [3:2] | nand_gpio_mux_d1 | R/W | Mux control of the NAND_D[1] and GPIO_1[29] pins<br>00: GPIO_1[29] pin is mux-ed out.<br>01: NAND_D[1] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |
| [1:0] | nand_gpio_mux_d0 | R/W | Mux control of the NAND_D[0] and GPIO_1[28] pins<br>00: GPIO_1[28] pin is mux-ed out.<br>01: NAND_D[0] pin is mux-ed out.<br>10: Reserved<br>11: Reserved | 0x0 | HR/WR |

### 4.4.2.18 Multi-function Port Setting Register 4 (Offset = 0x60)

**Table 4-21. Multi-function Port Setting Register 4 (MFPSR4)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | - | - | Reserved | - | - |
| [29:28] | txd3_gpio_mux | R/W | Mux control of the TXD3 and GPIO_2[31] pins<br>00: GPIO_2[31] pin is mux-ed out.<br>01: TXD3 pin is mux-ed out.<br>10: LCD_TV PCLK is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [27:26] | txd2_gpio_mux | R/W | Mux control of the TXD2 and GPIO_2[30] pins<br>00: GPIO_2[30] pin is mux-ed out.<br>01: TXD2 pin is mux-ed out.<br>10: The LCD_TV PCLK is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [25:24] | rxd3_gpio_mux | R/W | Mux control of the RXD3 and GPIO_2[29] pins<br>00: GPIO_2[29] pin is mux-ed out.<br>01: RXD3 pin is mux-ed out.<br>10: Reserved.<br>11: Reserved | 0x0 | HR/WR |
| [23:22] | rxd2_gpio_mux | R/W | Mux control of the RXD2 and GPIO_2[28] pins<br>00: GPIO_2[28] pin is mux-ed out.<br>01: RXD2 pin is mux-ed out. | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| | | | 10: Reserved | | |
| | | | 11: Reserved | | |
| [21:20] | ice_gpio_mux_6 | R/W | Mux control of the NTRST and GPIO_0[31] pins | 0x0 | HR/WR |
| | | | 00: NTRST pin is mux-ed in (CPU ICE). | | |
| | | | 01: GPIO_0[31] pin is mux-ed out. | | |
| | | | 10: Reserved. | | |
| | | | 11: IR-Det[16] is mux-ed in. | | |
| [19:18] | ice_gpio_mux_5 | R/W | Mux control of the TDI and GPIO_0[30] pins | 0x0 | HR/WR |
| | | | 00: TDI pin is mux-ed in (CPU ICE). | | |
| | | | 01: GPIO_0[30] pin is mux-ed out. | | |
| | | | 10: PWM[7] is mux-ed out. | | |
| | | | 11: IR-Det[15]/GPCN[15] is mux-ed in. | | |
| [17:16] | ice_gpio_mux_4 | R/W | Mux control of the TDO and GPIO_0[29] pins | 0x0 | HR/WR |
| | | | 00: TDO pin is mux-ed out (CPU ICE). | | |
| | | | 01: GPIO_0[29] pin is mux-ed out. | | |
| | | | 10: PWM[6] is mux-ed out. | | |
| | | | 11: IR-Det[14]/GPCN[14] is mux-ed in. | | |
| [15:14] | ice_gpio_mux_3 | R/W | Mux control of the TMS and GPIO_0[28] pins | 0x0 | HR/WR |
| | | | 00: TMS pin is mux-ed in (CPU ICE). | | |
| | | | 01: GPIO_0[28] pin is mux-ed in. | | |
| | | | 10: PWM[5] is mux-ed out. | | |
| | | | 11: IR-Det[13]/GPCN[13] is mux-ed in. | | |
| [13:12] | ice_gpio_mux_2 | R/W | Mux control of the TCK and GPIO_0[27] pins | 0x0 | HR/WR |
| | | | 00: TCK pin is mux-ed in (CPU ICE). | | |
| | | | 01: GPIO_0[27] pin is mux-ed in. | | |
| | | | 10: PWM[4] is mux-ed out. | | |
| | | | 11: IR-Det[12]/GPCN[12] is mux-ed in. | | |
| [11:9] | i2s_gpio_mux_txd | R/W | Mux control of the I2S_TXD and GPIO_0[22] pins | 0x0 | HR/WR |
| | | | 000: GPIO_0[22] pin is mux-ed out. | | |
| | | | 001: I2S_TXD pin is mux-ed out (I2S). | | |
| | | | 010: ct656[4] is mux-ed out. | | |
| | | | 011: UART4_SOUT | | |
| | | | 100: PWM[7] is mux-ed out. | | |
| | | | 101: SPI_FS5 pin is mux-ed out. | | |
| | | | 110: Reserved. | | |
| | | | 111: IR-Det[7]/GPCN[7] is mux-ed in. | | |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [8:6] | i2s_gpio_mux_clk | R/W | Mux control of the I2S_SCLK and the GPIO_0[21] pins<br>000: GPIO_0[21] pin is mux-ed out.<br>001: I2S_SCLK pin is mux-ed out (I2S).<br>011: UART4_SIN<br>100: PWM[6] is mux-ed out.<br>101: SPI_FS4 pin is mux-ed out.<br>110: Reserved<br>111: IR-Det[6]/GPCN[6] is mux-ed in. | 0x0 | HR/WR |
| [5:3] | i2s_gpio_mux_rxd | R/W | Mux control of the I2S_RXD and GPIO_0[20] pins<br>000: GPIO_0[20] pin is mux-ed out<br>001: I2S_RXD pin is mux-ed in (I2S)<br>010: ct656[6] is mux-ed out<br>011: UART3_SOUT<br>100: PWM[5] is mux-ed out<br>101: SPI_FS3 pin is mux-ed out<br>110: Reserved<br>111: IR-Det[5]/GPCN[5] is mux-ed in. | 0x0 | HR/WR |
| [2:0] | i2s_gpio_mux_fs | R/W | Mux control of the I2S_FS and GPIO_0[19] pins<br>000: GPIO_0[19] pin is mux-ed out.<br>001: I2S_FS pin is mux-ed out (I2S).<br>010: ct656[7] is mux-ed out.<br>011: UART3_SIN<br>100: PWM[4] is mux-ed out.<br>101: SPI_FS2 pin is mux-ed out.<br>110: Reserved.<br>111: IR-Det[4]/GPCN[4] is mux-ed in. | 0x0 | HR/WR |

### 4.4.2.19 Multi-function Port Setting Register 5 (Offset = 0x64)

**Table 4-22. Multi-function Port Setting Register 5 (MFPSR5)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | tvd15_gpio_mux | R/W | Mux control of the TV_D15 and GPIO_2[27] pins<br>00: GPIO_2[27] pin is mux-ed out.<br>01: MII_PDNPHY pin is mux-ed out.<br>10: TV_D15 is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [29:28] | tvd14_gpio_mux | R/W | Mux control of the TV_D14 and GPIO_2[26] pins<br>00: GPIO_2[26] pin is mux-ed out.<br>01: MII_PHYLINK pin is mux-ed out.<br>10: TV_D14 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [27:26] | tvd13_gpio_mux | R/W | Mux control of the TV_D13 and GPIO_2[25] pins<br>00: GPIO_2[25] pin is mux-ed out.<br>01: MII_MDIO is mux-ed out.<br>10: TV_D13 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [25:24] | tvd12_gpio_mux | R/W | Mux control of the TV_D12 and GPIO_2[24] pins<br>00: GPIO_2[24] pin is mux-ed out.<br>01: MII_MDC pin is mux-ed out.<br>10: TV_D12 is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [23:22] | tvd11_gpio_mux | R/W | Mux control of the TV_D11 and GPIO_2[23] pins<br>00: GPIO_2[23] pin is mux-ed out.<br>01: MII_CLKOUT pin is mux-ed out.<br>10: TV_D11 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [21:20] | tvd10_gpio_mux | R/W | Mux control of the TV_D10 and GPIO_2[22] pins<br>00: GPIO_2[22] pin is mux-ed out.<br>01: MII_COL pin is mux-ed out.<br>10: TV_D10 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [19:18] | tvd9_gpio_mux | R/W | Mux control of the TV_D9 and GPIO_2[21] pins<br>00: GPIO_2[21] pin is mux-ed out.<br>01:MII_TXEN pin is mux-ed out.<br>10: TV_D9 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [17:16] | tvd8_gpio_mux | R/W | Mux control of the TV_D8 and GPIO_2[20] pins<br>00: GPIO_2[20] pin is mux-ed out.<br>01: MII_TXD[1] pin is mux-ed out.<br>10: TV_D8 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [15:14] | tvd7_gpio_mux | R/W | Mux control of the TV_D7 and GPIO_2[19] pins<br>00: GPIO_2[19] pin is mux-ed out.<br>01: MII_TXD[0] pin is mux-ed out.<br>10: TV_D7 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [13:12] | tvd6_gpio_mux | R/W | Mux control of the TV_D6 and GPIO_2[18] pins<br>00: GPIO_2[18] pin is mux-ed out.<br>01: MII_TXCK pin is mux-ed out.<br>10: TV_D6 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [11:10] | tvd5_gpio_mux | R/W | Mux control of the TV_D5 and GPIO_2[17] pins<br>00: GPIO_2[17] pin is mux-ed out.<br>01: MII_RXCK pin is mux-ed out.<br>10: TV_D5 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [9:8] | tvd4_gpio_mux | R/W | Mux control of the TV_D4 and GPIO_2[16] pins<br>00: GPIO_2[16] pin is mux-ed out.<br>01: MII_RXD[1] pin is mux-ed out.<br>10: TV_D4 data[4] pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [7:6] | tvd5_gpio_mux | R/W | Mux control of the TV_D3 and GPIO_2[15] pins<br>00: GPIO_2[15] pin is mux-ed out.<br>01: MII_RXD[0] pin is mux-ed out.<br>10: TV_D3 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [5:4] | tvd2_gpio_mux | R/W | Mux control of the TV_D2 and GPIO_2[10] pins<br>00: GPIO_2[10] pin is mux-ed out.<br>01: MII_RXDV pin is mux-ed out.<br>10: TV_D2 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [3:2] | tvd1_gpio_mux | R/W | Mux control of the TV_D1and GPIO_0[26] pins<br>00: GPIO_0[26] pin is mux-ed out.<br>01: MII_RXCRS pin is mux-ed out.<br>10: TV_D1 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |
| [1:0] | tvd0_gpio_mux | R/W | Mux control of the TV_D0 and GPIO_0[25] pins<br>00: GPIO_0[25] pin is mux-ed out.<br>01: MII_RXER pin is mux-ed out.<br>10: TV_D0 pin is mux-ed out.<br>11: Reserved | 0x0 | HR/WR |

### 4.4.2.20 System Controller Control Register (Offset = 0x6C)

**Table 4-23. System Controller Control Register (SCCR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:27] | - | - | Reserved | - | - |
| 26 | SSP1_sel | R/W | SSP1 interface selection<br>0: Select the external interface<br>1: Select the internal interface | 0x0 | HR/WR |
| [25:24] | video-in pin mux | R/W | Video-in pin mux<br>0: Original interface<br>1: Mirror interface<br>2: Swap interface<br>3: Mirror+Swap interface | 0x0 | HR/WR |
| 23 | I2C_sel | R/W | $I^2C$ interface selection<br>0: Select the $I^2C$ interface<br>1: Select the CAP1 interface | 0x0 | HR/WR |
| 22 | nand_sd _sel | R/W | NAND/SD bus interface selection<br>0: Select the NAND bus interface<br>1: Select the SD card interface | 0x0 | HR/WR |
| 21 | SD_up | R/W | SD ctrl pull-up control<br>0: No pull-up the control pin<br>1: Pull-up the control pin | 0x0 | HR/WR |
| 20 | NAND_up | R/W | NAND interface pull-up control<br>0: No pull-up the NAND interface<br>1: Pull-up the NAND interface | 0x0 | HR/WR |
| 19 | RMII_sel | R/W | MII/RMII interface selection<br>0: Select MII interface<br>1: Select RMII interface | 0x0 | HR/WR |
| 18 | Regulator enable | R/W | Regulator enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |
| 17 | CMOS clock enable | R/W | cmos_clkout output enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 16 | TV output enable | R/W | ct.656 data/clk output enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |
| 15 | extclk3 enable | R/W | extclk3 output enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |
| 14 | extclk2 enable | R/W | extclk2 output enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |
| 13 | extclk1 enable | R/W | extclk1 output enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |
| 12 | extclk0 enable | R/W | extclk0 output enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |
| 11 | mac phy bypass | R/W | MAC PHY bypass control<br>0: Normal<br>1: Bypass | 0x0 | HR/WR |
| 10 | mac clock enable | R/W | mii_txck/mii_clk output enable control<br>0: Disable<br>1: Enable | 0x0 | HR/WR |
| 9 | AES resetn | R/W | Extra reset control for the AES module<br>0: Reset AES<br>1: Release AES | 0x0 | HR/WR |
| 8 | mcp210 resetn | R/W | Extra reset control for the H.264 decoder<br>0: Reset H.264 encoder<br>1: Release H.264 encoder | 0x0 | HR/WR |
| [7:4] | edge-sync DDR select | R/W | Select delay time of the edgesyn signal on internal DDRII<br>0000: Minimal delay<br>---<br>1111: Maximal delay | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [3:0] | edge-sync CPU select | R/W | Select delay time of the edgesyn signal on internal CPU<br>0000: Minimal delay<br>---<br>1111: Maximal delay | 0x0 | HR/WR |

### 4.4.2.21 Divider Setting Register 0 (Offset = 0x70)

**Table 4-24. Divider Setting Register 0 (DSR0)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | - | - | Reserved | - | - |
| [29:24] | CMOSCLK_PVALUE | R/W | Select the CMOS CLK frequency<br>CMOS_CLK = PLL2_output/ (CMOSCLK_PVALUE + 1) | 0x13 | HR/WR |
| [23:18] | EXTCLK3_PVALUE | R/W | Select the EXT3 CLK frequency<br>EXT_CLK3 = PLL2_output/ (EXTCLK3_PVALUE + 1) | 0x4 | HR/WR |
| [17:12] | EXTCLK2_PVALUE | R/W | Select the EXT2 CLK frequency<br>EXT_CLK2 = PLL2_output/ (EXTCLK2_PVALUE + 1) | 0x4 | HR/WR |
| [11:6] | EXTCLK1_PVALUE | R/W | Select the EXT1 CLK frequency<br>EXT_CLK1 = PLL2_output/ (EXTCLK1_PVALUE + 1) | 0x4 | HR/WR |
| [5:0] | EXTCLK0_PVALUE | R/W | Select the EXT0 CLK frequency<br>EXT_CLK0 = PLL2_output/ (EXTCLK0_PVALUE + 1) | 0x4 | HR/WR |

### 4.4.2.22 Divider Setting Register 1 (Offset = 0x74)

**Table 4-25. Divider Setting Register 1 (DSR1)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:26] | MACCLK_PVALUE | R/W | Select the MACCLK frequency<br>MAC_CLK = PLL1_output/ (MACCLK_PVALUE + 1) | 0x7 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [25:20] | LCD_PVALUE | R/W | Select the LCD CLK frequency<br>LCD_CLK = PLL2_output/ (LCD_PVALUE + 1) | 0x13 | HR/WR |
| [19:16] | SCARCLK_PVALUE | R/W | Select the SCALAR CLK frequency<br>SCALAR_CLK = PLL2_output/ (SCARCLK_PVALUE + 1) | 0x3 | HR/WR |
| [15:12] | - | - | Reserved | - | - |
| [11:8] | LCD_S_PVALUE | R/W | Select the scaler clock of LCD controller<br>LC_SCALER_CLK = PLL2_output/ (LCD_S_PVALUE + 1) | 0x3 | HR/WR |
| [7:4] | PWMCLK_PVALUE | R/W | Select the PWM frequency<br>PWM_CLK = PLL2_output/ (PWMCLK_PVALUE + 1) | 0x3 | HR/WR |
| [3:0] | SDCCLK_PVALUE | R/W | Select the SDC CLK frequency<br>SDC_CLK = PLL1_output/ (SDCCLK_PVALUE + 1) | 0x3 | HR/WR |

### 4.4.2.23 Divider Setting Register 2 (Offset = 0x78)

**Table 4-26. Divider Setting Register 2 (DSR2)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:30] | - | - | Reserved | - | - |
| [29:24] | AUDIOCLK_PVALUE | R/W | Select the AUDIO CLK frequency<br>AUDIO_CLK = PLL2_output/ (AUDIOCLK_PVALUE + 1) | 0x2c | HR/WR |
| [23:18] | UARTCLK_PVALUE | R/W | Select the UART CLK frequency<br>UART_CLK = PLL2_output/ (UARTCLK_PVALUE + 1) | 0xf | HR/WR |
| [17:12] | - | - | Reserved | - | - |
| [11:6] | SSP1CLK_PVALUE | R/W | Select the ssp1 CLK frequency<br>SSP1CLK = PLL2_output/ (SSP1CLK_PVALUE + 1) | 0x15 | HR/WR |
| [5:0] | SSP0CLK_PVALUE | R/W | Select the ssp0 CLK frequency<br>SSP0CLK = PLL2_output/ (SSP0CLK_PVALUE + 1) | 0x15 | HR/WR |

### 4.4.2.24 TVE DAC Control/ Divider Setting Register3 (Offset = 0x7C)

**Table 4-27. TVE DAC Control/Divider Setting Register 3 (DSR3)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:26] | ADCCLK_PVALUE | R/W | Select the ADC CLK frequency<br>ADC CLK = PLL2_output/ (ADCCLK_PVALUE + 1) | 0x27 | HR/WR |
| [25:21] | MCP100CLK_PVALUE | R/W | Select the MCP100 CLK frequency<br>MCP100_CLK = PLL1_output/ (MCP100CLK_PVALUE + 1) | 0x3 | HR/WR |
| [20:16] | MCP210CLK_PVALUE | R/W | Select the MCP210 CLK frequency<br>MCP210 CLK = PLL1_output/ (MCP210CLK_PVALUE + 1) | 0x3 | HR/WR |
| [15:12] | ISPSENCLK_PVALUE | R/W | Select the ISP/VCAP0/VCAP1 MCLK frequency<br>ISPSEN CLK = PLL1_output/ (ISPSENCLK_PVALUE + 1) | 0x3 | HR/WR |
| [11:4] | - | - | Reserved | - | - |
| 3 | tve_dac_stb | R/W | Standby mode control<br>0: Normal operation<br>1: Standby mode operation | 0x1 | HR/WR |
| 2 | tve_dac_pd | R/W | Power-down mode control<br>0: Normal operation<br>1: Power-down mode operation | 0x0 | HR/WR |
| 1 | tve_dac_iso_en | R/W | Isolated mode control<br>0: Normal operation<br>1: Isolated mode operation | 0x0 | HR/WR |
| 0 | tve_dac_psw_pd | R/W | Power switch control of VCC10K<br>0: Normal operation<br>1: Power-down mode | 0x0 | HR/WR |

### 4.4.2.25 DDR2 PHY Control Register (Offset = 0x80)

**Table 4-28. DDR2 PHY Control Register (DPCR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:4] | - | - | Reserved | - | - |
| 3 | ddr2_phy_fldo_data | R/W | - | 0x0 | HR/WR |
| 2 | - | - | Reserved | - | - |
| 1 | ddr2_phy_rst_n | R/W | This bit resets the DDR2 PHY block.<br>Active low | 0x0 | HR/WR |
| 0 | ddr2_phy_pdn | R/W | This bit controls the DDR PHY DLL power-down mode.<br>0: Power-down mode<br>1: Normal operation | 0x0 | HR/WR |

### 4.4.2.26 MAC PHY Control Register (Offset = 0x84)

**Table 4-29. MAC PHY Control Register (MPCR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 31 | mac_phy_cableoff | R/W | MAC PHY cable off | 0x0 | HR/WR |
| [30:21] | - | R/W | - | - | - |
| 20 | mac_phy_vhopa | R/W | Fine tune the TX 100M output waveform for the overshoot specification | 0x0 | HR/WR |
| [19:17] | - | - | Reserved | - | - |
| 16 | mac_phy_pwrsv | R/W | This bit is used for the internal test<br>In the normal function mode, please tie to 0. | 0x0 | HR/WR |
| [15:14] | mac_phy_refsel | R/W | Speed change for OSC<br>00: 25 MHz (Off-chip crystal)<br>01: 50 MHz (Off-chip crystal)<br>10: 75 MHz<br>11: 100 MHz | 0x0 | HR/WR |
| [13:8] | mac_phy_addr | R/W | PHY address | 0x1 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 7 | mac_phy_clksrc | R/W | Select the crystal clock input source or the chip clock input source<br>0: Choose the on-chip clock from the PLL1 source<br>1: Choose the crystal clock input source | 0x0 | HR/WR |
| [6:4] | opmode | R/W | Operation mode setting<br>In the normal function mode, tie these bits to low. | 0x0 | HR/WR |
| 3 | loopback | R/W | MAC PHY loopback enable<br>0: Disable the loopback mode<br>1: Enable the loopback mode | 0x0 | HR/WR |
| 2 | mac_phy_oscmd | R/W | This signal controls the crystal-alive mode. | 0x0 | HR/WR |
| 1 | mac_phy_pdn | R/W | The MAC PHY internal unction power-down control | 0x0 | HR/WR |
| 0 | mac_phy_rstn | R/W | The MAC PHY hardware reset should be longer than 500 ns. | 0x1 | HR/WR |

### 4.4.2.27 MAC PHY ID Register (Offset = 0x88)

**Table 4-30. MAC PHY ID Register (MPIDR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:0] | MAC PHY ID | R/W | MAC PHY identifier | 0x0 | HR/WR |

### 4.4.2.28 Audio Codec Control Register 0 (Offset = 0x8C)

**Table 4-31. Audio Codec Control Register 0 (ACR0)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:16] | - | - | Reserved | - | - |
| [15:12] | MCLK DAC Mode | R/W | Define the ratio of MCLK frequency and DAC sampling rate<br>1111: MCLK = 256 x sampling clock<br>1110: MCLK = 128 x sampling clock<br>1101: MCLK = 272 x sampling clock<br>1100: MCLK = 136 x sampling clock | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| | | | 1011: MCLK = 1024 x sampling clock | | |
| | | | 1010: MCLK = 512 x sampling clock | | |
| | | | 1001: MCLK = 1088 x sampling clock | | |
| | | | 1000: MCLK = 544 x sampling clock | | |
| | | | 0111: MCLK = 384 x sampling clock | | |
| | | | 0110: MCLK = 250 x sampling clock | | |
| | | | 0101: MCLK = 500 x sampling clock | | |
| | | | 0100: MCLK = 1500 x sampling clock | | |
| | | | 0011: MCLK = 1536 x sampling clock | | |
| | | | 0010: MCLK = 768 x sampling clock | | |
| | | | 0001: MCLK = 750 x sampling clock | | |
| | | | 0000: MCLK = 375 x sampling clock | | |
| [11:10] | DAIMODE | R/W | Digital audio interface mode | 0x0 | HR/WR |
| | | | 00: Parallel data mode | | |
| | | | 01: $I^2S$ serial data mode | | |
| | | | 10: Right-justified serial data mode (For DAC) | | |
| | | | 11: Left-justified serial data mode (For DAC) | | |
| [9:8] | DEEMPH | R/W | DAC digital de-emphasis filter control signal | 0x0 | HR/WR |
| | | | 11: Set 48 kHz de-emphasis filter on with MCLK = 256x | | |
| | | | 10: Set 44.1 kHz de-emphasis filter on with MCLK = 256x | | |
| | | | 01: Set 32 kHz de-emphasis filter on with MCLK = 384x | | |
| | | | 00: Set the de-emphasis filter off | | |
| [7:4] | MCLK ADC Mode | R/W | Define the ratio of MCLK frequency and ADC sampling rate | 0x0 | HR/WR |
| | | | 1111: MCLK = 256 x sampling clock | | |
| | | | 1110: MCLK = 128 x sampling clock | | |
| | | | 1101: MCLK = 272 x sampling clock | | |
| | | | 1100: MCLK = 136 x sampling clock | | |
| | | | 1011: MCLK = 1024 x sampling clock | | |
| | | | 1010: MCLK = 512 x sampling clock | | |
| | | | 1001: MCLK = 1088 x sampling clock | | |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| | | | 1000: MCLK = 544 x sampling clock | | |
| | | | 0111: MCLK = 384 x sampling clock | | |
| | | | 0110: MCLK = 250 x sampling clock | | |
| | | | 0101: MCLK = 500 x sampling clock | | |
| | | | 0100: MCLK = 1500 x sampling clock | | |
| | | | 0011: MCLK = 1536 x sampling clock | | |
| | | | 0010: MCLK = 768 x sampling clock | | |
| | | | 0001: MCLK = 750 x sampling clock | | |
| | | | 0000: MCLK = 375 x sampling clock | | |
| 3 | REBACK | R/W | The DAC input digital data is back to the ADC digital output.<br>In the serial $I^2S$ mode:<br>1: Enable the reback function<br>0: Normal mode | 0x0 | HR/WR |
| 2 | TESTPD | R/W | DAC anti-pop sound scheme control<br>0: Enable the anti-pop sound mechanism<br>1: Disable the anti-pop sound mechanism | 0x0 | HR/WR |
| 1 | MASTER | R/W | Audio interface control<br>0: Slave mode<br>1: Master mode | 0x0 | HR/WR |
| 0 | ADDA_RESET | R/W | The bit resets the ADDA audio codec<br>Active high | 0x0 | HR/WR |

### 4.4.2.29 Audio Codec ADC Control Register 1 (Offset = 0x90)

**Table 4-32. Audio Codec ADC Control Register 1 (ACR1)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:29] | ALCNGTH | R/W | ALC noise gate threshold<br>111 = -36 dB<br>110 = -42 dB<br>101 = -48 dB<br>(6 dB per step)<br>000 = -78 dB | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 28 | ADCDWA | R/W | ADC modulator DWA mode control (When ADCDEM = '0x0', this register can be controlled.)<br>1: Enabled<br>0: Disabled | 0x0 | HR/WR |
| 27 | ADCDEM | R/W | ADC modulator DEM mode control<br>1: Enabled<br>0: Disabled | 0x0 | HR/WR |
| 26 | ALC | R/W | ALC function select<br>1: Enabled<br>0: Disabled | 0x0 | HR/WR |
| 25 | HPF | R/W | ADC high pass filter control<br>1: Enabled<br>0: Disabled | 0x0 | HR/WR |
| [24:23] | ADCGAIN | R/W | ADC digital modulator gain setting in the test mode<br>00: +1 dB<br>01: +1.5 dB<br>10: +2 dB<br>11: +4 dB (Default) | 0x3 | HR/WR |
| 22 | LADCPD | R/W | ADC power-down control of the left channel<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |
| 21 | RADCPD | R/W | ADC power-down control of the right channel<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |
| 20 | LFEPD | R/W | ADC front-end power-down control of the left channel<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |
| 19 | RFEPD | R/W | ADC front-end power-down control of the right channel<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [18:13] | LIV | R/W | ADC gain control of the left channel<br>6-bit line input volume control of the left channel<br>111111: +36 dB<br>000000: -27 dB<br>There are a total of 64 steps, each step is 1 dB resolution.<br>For 0 dB, LIV[5:0] = 011011 | 0x0 | HR/WR |
| [12:7] | RIV | R/W | ADC gain control of the right channel<br>6-bit line input volume control of the right channel<br>111111: +36 dB<br>000000: -27 dB<br>There are a total of 64 steps, each step is 1 dB resolution.<br>For 0 dB, LIV[5:0] = 011011 | 0x0 | HR/WR |
| 6 | LIM | R/W | ADC input mute of the left channel<br>1: Enable input mute<br>0: Normal operation | 0x0 | HR/WR |
| 5 | RIM | R/W | ADC input mute of the right channel<br>1: Enable input mute<br>0: Normal operation | 0x0 | HR/WR |
| 4 | LMICBT | R/W | LMICIN boost control<br>1: +20 dB gain<br>0: 0 dB gain | 0x0 | HR/WR |
| 3 | RMICBT | R/W | RMICIN boost control<br>1: +20 dB gain<br>0: 0 dB gain | 0x0 | HR/WR |
| 2 | LINSEL | R/W | ADC input signal selection of the left channel<br>1: Select LMICIN<br>0: Select LLINEIN | 0x0 | HR/WR |
| 1 | RINSEL | R/W | ADC input signal selection of the right channel<br>1: Select RMICIN<br>0: Select RLINEIN | 0x0 | HR/WR |
| 0 | CPS | R/W | Chopper stabilized control<br>1: Chopper stabilized on<br>0: Chopper stabilized off | 0x0 | HR/WR |

### 4.4.2.30 Audio Codec Control Register 2 (Offset = 0x94)

**Table 4-33. Audio Codec Control Register 2 (ACR2)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:28] | ALCDCYT | R/W | Set the ALC decay time (1 dB per step)<br>0000 = 104 µs<br>0001 = 208 µs<br>0010 = 416 µs<br>(Time is doubled with every step.) | 0x0 | HR/WR |
| [27:24] | ALCATKT | R/W | Set the ALC attack time (1 dB per step)<br>0000 = 63.3 µs<br>0001 = 125 µs<br>0010 = 250 µs<br>(Time is doubled with every step.) | 0x0 | HR/WR |
| [23:20] | ALCTL | R/W | Set the ALC target level<br>1111 = -4 dBFS<br>1110 = -6 dBFS<br>(2 dB per step)<br>0000 = -34 dBFS | 0x0 | HR/WR |
| [19:17] | ALCMIN | R/W | Set the PGA minimum gain<br>111 = -6 dB<br>110 = -9 dB<br>101 = -12 dB<br>100 = -15 dB<br>011 = -18 dB<br>010 = -21 dB<br>001 = -24 dB<br>000 = -27 dB | 0x0 | HR/WR |
| [16:14] | ALCMAX | R/W | Set the PGA maximum gain<br>111 = +36 dB<br>110 = +30 dB<br>101 = +24 dB<br>100 = +18 dB<br>011 = +12 dB<br>010 = +6 dB<br>001 = +0 dB<br>000 = -6 dB | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [11:6] | LDAV | R/W | DAC digital volume gain control of the left channel<br>1111111: +30 dB<br>0101111: -50 dB (Minimum gain)<br>Each step is 1 dB resolution.<br>For 0 dB, LADV[6:0] = 1100001 | 0x0 | HR/WR |
| [5:0] | RDAV | R/W | 7-bit ADC digital volume gain control of the right channel<br>1111111: +30 dB<br>0101111: -50 dB (Minimum gain)<br>Each step is 1 db resolution.<br>For 0 dB, LADV[6:0] = 1100001 | 0x0 | HR/WR |

### 4.4.2.31 Audio Codec Control Register 3 (Offset = 0x98)

**Table 4-34. Audio Codec Control Register 3 (ACR3)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:28] | ALCHOLT | R/W | Set the ALC hold time before gain increased<br>0000 = 0 µs<br>0001 = 250 µs<br>0010 = 500 µs<br>0011 = 1 ms<br>(Time is doubled with every step) | 0x0 | HR/WR |
| 27 | ALCZC | R/W | ALC update gain on zero-cross function control<br>1: Enable<br>0: Disable | 0x0 | HR/WR |
| 26 | LBYPSEL | R/W | LLINEIN bypass to LHPOUT control<br>1: Enable the bypass control<br>0: Disable the bypass control | 0x0 | HR/WR |
| 25 | RBYPSEL | R/W | RLINEIN bypass to RHPOUT control<br>1: Enable the bypass control<br>0: Disable the bypass control | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 24 | LHPSEL | R/W | SCF DAC to RHPOUT control of the left channel<br>1: Enable SCF DAC to LHPOUT control<br>0: Disable SCF DAC to LHPOUT control | 0x0 | HR/WR |
| 23 | RHPSEL | R/W | SCF DAC to RHPOUT control of the right channel<br>1: Enable SCF DAC to LHPOUT control<br>0: Disable SCF DAC to LHPOUT control | 0x0 | HR/WR |
| 22 | LDACPD | R/W | DAC power-down control of the left channel<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |
| 21 | RDACPD | R/W | DAC power-down control of the right channel<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |
| [20:19] | DAMIXINV | R/W | Set DAC channel mix inverter function when DAMIXER = '1'<br>00: Normal mode<br>01: Left-channel inverter<br>10: Right-channel inverter<br>11: Left-channel and Right-channel inverters | 0x0 | HR/WR |
| 18 | DAMIXER | R/W | DAC mix control of the left and right channels<br>1: Enabled<br>0: Disabled | 0x0 | HR/WR |
| 17 | LHM | R/W | DAC digital mute of the left channel<br>The left SCF DAC is mute.<br>1: Enable<br>0: Disable | 0x0 | HR/WR |
| 16 | RHM | R/W | DAC digital mute of the right channel<br>The right SCF DAC is mute.<br>1: Enable<br>0: Disable | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 15 | SDMDWA | R/W | DAC modulator DWA mode control (When SDMDEM = '0', this bit is valid.)<br>1: Enable<br>0: Disable | 0x0 | HR/WR |
| 14 | SDMDEM | R/W | DAC modulator DEM mode control<br>1: Enable<br>0: Disable | 0x0 | HR/WR |
| [13:12] | SDMGAIN | R/W | DAC digital modulator gain attenuation in the test mode<br>00: -1 dB<br>01: -2 dB (Default)<br>10: -3 dB<br>11: -6 dB | 0x0 | HR/WR |
| [11:6] | LDAV | R/W | 6-bit DAC digital volume control of the left channel<br>111111: 0 dB<br>100011: -40 dB<br>There are a total of 41 steps, and each step is 1 dB resolution<br>For values smaller than 100011, the gain is set to -40 dB. | 0x0 | HR/WR |
| [5:0] | RDAV | R/W | 6-bit DAC digital volume control of the right channel<br>111111: 0 dB<br>100011: -40 dB<br>There are a total of 41 steps, and each step is 1 dB resolution<br>For values smaller than 100011, the gain is set to -40 dB. | 0x0 | HR/WR |

### 4.4.2.32 Audio Codec DAC Control Register 4 (Offset = 0x9C)

**Table 4-35. Audio Codec Control Register 4 (ACR4)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:29] | - | - | Reserved | - | - |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 28 | MICSEL | R/W | ADC digital microphone control<br>1: Enable the digital microphone<br>0: Disable the digital microphone | | |
| [27:26] | IRSEL | R/W | Reference current control<br>11: Normal setting<br>10: Reserved<br>01: Reserved<br>00: Reserved | 0x0 | HR/WR |
| 25 | MICPD | R/W | Microphone bias power-down control<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |
| 24 | SPK_PD | R/W | Power-down control for the speaker driver<br>1: Power-down mode<br>0: Normal mode | 0x0 | HR/WR |
| [23:22] | SPKSEL | R/W | SCF DAC to speaker control<br>00: Mute<br>01: Reserved<br>10: DAMIXER = 1,<br>DAMIXINV[1:0] = 00<br>SCF DAC to BTL speaker is enabled.<br>11: Reserved | 0x0 | HR/WR |
| [21:16] | SPV | R/W | 6-bit speaker volume control<br>111111: + 6 dB<br>010001: - 40 dB | 0x0 | HR/WR |
| 15 | DEPOP | R/W | Pop sound control (Test only)<br>1: Enable<br>0: Disable | 0x0 | HR/WR |
| 14 | VCMHP_PD | R/W | Power-down control of the headphone VCMD driver<br>1: Power down the headphone VCM driver<br>0: Normal mode | 0x0 | HR/WR |
| 13 | LHPPD | R/W | Power-down control of the left-channel headphone driver<br>1: Power down the left channel headphone driver<br>0: Normal mode | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| 12 | RHPPD | R/W | Power-down control of the right-channel headphone driver<br>1: Power down the right channel headphone driver<br>0: Normal mode | 0x0 | HR/WR |
| [11:6] | LHV | R/W | 6-bit headphone volume control of the left channel<br>Reserved | 0x0 | HR/WR |
| [5:0] | RHV | R/W | 6-bit headphone volume control of the right channel<br>Reserved | 0x0 | HR/WR |

### 4.4.2.33 OTG Control Register (Offset = 0xA0)

**Table 4-36. OTG Control Register 1 (OTGCR)**

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [31:20] | - | R/W | Reserved | - | - |
| [19:17] | otg_phy_tcba | R/W | 3-bit signal to control the reference voltage in the squelch circuit | 0x4 | HR/WR |
| [16:12] | otg_phy_vctrl | R/W | 5-bit vendor control | 0x1 | HR/WR |
| 11 | otg_phy_pllalive | R/W | Power-On Reset (POR) signal<br>0: PLL is not alive.<br>1: PLL is alive. | 0x0 | HR/WR |
| 10 | - | R/W | Reserved | - | - |
| 9 | otg_phy_osc_en | R/W | CLKOSCO clock output selection<br>0: CLKOSCO is enabled.<br>1: CLKOSCO is disabled. | 0x0 | HR/WR |
| 8 | otg_phy xtlsel | R/W | Crystal source selection<br>0: 12-MHz crystal input<br>1: 30-MHz crystal input | 0x0 | HR/WR |
| [7:4] | - | R/W | Reserved | - | - |
| 3 | u_iddig | R/W | ID Dig<br>This signal indicates that the connected plug is either a mini-A or mini-B. It must be valid within 20 ms after u_idpulup is set to 1b.<br>0: Connected plug is a mini-A.<br>1: Connected plug is a mini-B. | 0x0 | HR/WR |

| Bit | Name | Type | Description | Reset Value | Reset Type |
|---|---|---|---|---|---|
| [2:1] | - | R/W | Reserved | - | - |
| 0 | u_vbusvld | R/W | - | 0x1 | HR/WR |

The following tables describe the detailed pin-mux function.

**Table 4-37. Pin-mux Table 1**

| spi_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| SPI_FS0 | GPIO_0[14] | SPI_FS0 | X | X |
| SPI_FS1 | GPIO_0[15] | SPI_FS1 | X | X |
| SPI_RXD | GPIO_0[16] | SPI_RXD | X | X |
| SPI_SCLK | GPIO_0[17] | SPI_SCLK | X | X |
| SPI_TXD | GPIO_0[18] | SPI_TXD | X | X |

**Table 4-38. Pin-mux Table 2**

| UART_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| UART0_SIN | UART0_SIN | GPIO_0[4] | X | X |
| UART0_SOUT | UART0_SOUT | GPIO_0[5] | X | X |
| UART1_SIN | GPIO_0[6] | UART1_SIN | X | X |
| UART1_SOUT | GPIO_0[7] | UART1_SOUT | X | X |
| UART2_SIN | GPIO_0[8] | UART2_SIN | UART1_CTSN | NAND_BUSYn |
| UART2_SOUT | GPIO_0[9] | UART2_SOUT | UART1_RTSN | UART1_DTRN |
| UART3_SIN | GPIO_0[10] | UART3_SIN | UART1_DSM | UART2_DTRN |
| UART3_SOUT | GPIO_0[11] | UART3_SOUT | UART1_DTRN | UART1_DTRN |
| UART4_SIN | GPIO_0[12] | UART4_SIN | UART_RIN | DMIC_DATA |
| UART4_SOUT | GPIO_0[13] | UART4_SOUT | UART1_DCDN | DMIC_CLK |

**Table 4-39. Pin-mux Table 3**

| pwm_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| PWM[0] | GPIO_0[0] | PWM[0] | EXT_CLK[0] | DMIC_DATA |
| PWM[1] | GPIO_0[1] | PWM[1] | EXT_CLK[1] | DMIC_CLK |
| PWM[2] | GPIO_0[2] | PWM[2] | EXT_CLK[2] | IR-Det[10]/GPCN[10] |
| PWM[3] | GPIO_0[3] | PWM[3] | EXT_CLK[3] | IR-Det[11]/GPCN[11] |

**Table 4-40. Pin-mux Table 4**

| cap0_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| CAP0_FLD | GPIO_1[0] | CAP0_FLD | Bayer_strobe | X |
| CAP0_HS | GPIO_1[1] | CAP0_HS | Bayer_line | X |
| CAP0_VS | GPIO_1[2] | CAP0_VS | Bayer_frame | X |
| CAP0_CLK | GPIO_1[3] | CAP0_CLK | Bayer_clk | X |
| CAP0_D0 | GPIO_1[4] | CAP0_D0 | Bayer_D4 | X |
| CAP0_D1 | GPIO_1[5] | CAP0_D1 | Bayer_D5 | X |
| CAP0_D2 | GPIO_1[6] | CAP0_D2 | Bayer_D6 | X |
| CAP0_D3 | GPIO_1[7] | CAP0_D3 | Bayer_D7 | X |
| CAP0_D4 | GPIO_1[8] | CAP0_D4 | Bayer_D8 | X |
| CAP0_D5 | GPIO_1[9] | CAP0_D5 | Bayer_D9 | X |
| CAP0_D6 | GPIO_1[10] | CAP0_D6 | Bayer_D10 | X |
| CAP0_D7 | GPIO_1[11] | CAP0_D7 | Bayer_D11 | X |

**Table 4-41. Pin-mux Table 4**

| cap1_mux | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| CAP_CLKOUT | GPIO_1[12] | CAP_CLKOUT | ISP_CLKOUT | X | X | X | | |
| CAP1_CLK | GPIO_1[13] | CAP1_CLK | CT656_CLKOUT | X | X | X | | |
| CAP1_D0 | GPIO_1[14] | CAP1_D0 | CT656_D0 | X | PWM[0] | IR-Det[0]/GPCN[0] | SSP0_SCLK | |
| CAP1_D1 | GPIO_1[15] | CAP1_D1 | CT656_D1 | X | PWM[1] | IR-Det[1]/GPCN[1] | SSP0_TXD | |
| CAP1_D2 | GPIO_1[16] | CAP1_D2 | CT656_D2 | X | PWM[2] | IR-Det[2]/GPCN[2] | SSP0_RXD | I2S_SDA |
| CAP1_D3 | GPIO_1[17] | CAP1_D3 | CT656_D3 | X | PWM[3] | IR-Det[3]/GPCN[3] | SSP0_FS6 | I2S_SCL |
| CAP1_D4 | GPIO_1[18] | CAP1_D4 | Bayer_D0 | X | X | X | | |
| CAP1_D5 | GPIO_1[19] | CAP1_D5 | Bayer_D1 | X | X | X | | |
| CAP1_D6 | GPIO_1[20] | CAP1_D6 | Bayer_D2 | X | X | X | | |
| CAP1_D7 | GPIO_1[21] | CAP1_D7 | Bayer_D3 | X | X | X | | |

**Table 4-42. Pin-mux Table 5**

| ice_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| NTRST | NTRST | GPIO_0[31] | PWM[4] | IR-Det[12]/GPCN[12] |
| TCK | TCK | GPIO_0[30] | PWM[5] | IR-Det[13]/GPCN[13] |
| TDI | TDI | GPIO_0[29] | PWM[6] | IR-Det[14]/GPCN[14] |
| TDO | TDO | GPIO_0[28] | PWM[7] | IR-Det[15]/GPCN[15] |
| TMS | TMS | GPIO_0[27] | | IR-Det[16] |

**Table 4-43. Pin-mux Table 6**

| sd_mux | 0 | 1 |
|---|---|---|
| SD_CD | SD_CD | GPIO_1[27] |
| SD_CLK | SD_CLK | GPIO_1[26] |
| SD_CMD | SD_CMD | GPIO_1[25] |

**Table 4-44. Pin-mux Table 7**

| I2S_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| I2S_FS | GPIO_0[19] | I2S_FS | CT656_D7 | UART3_SIN |
| I2S_RXD | GPIO_0[20] | I2S_RXD | CT656_D6 | UART3_SOUT |
| I2S_SCLK | GPIO_0[21] | I2S_SCLK | CT656_D5 | UART4_SIN |
| I2S_TXD | GPIO_0[22] | I2S_TXD | CT656_D4 | UART4_SOUT |
| **I2S_mux** | **4** | **5** | **6** | **7** |
| I2S_FS | PWM[4] | SPI_FS2 | X | IR-Det[4]/GPCN[4] |
| I2S_RXD | PWM[5] | SPI_FS3 | X | IR-Det[5]/GPCN[5] |
| I2S_SCLK | PWM[6] | SPI_FS4 | X | IR-Det[6]/GPCN[6] |
| I2S_TXD | PWM[7] | SPI_FS5 | X | IR-Det[7]/GPCN[7] |

**Table 4-45. Pin-mux Table 8**

| I2S_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| NAND_D0 | GPIO_1[28] | NAND_D0 | X | X |
| NAND_D1 | GPIO_1[29] | NAND_D1 | X | X |
| NAND_D2 | GPIO_1[30] | NAND_D2 | X | X |
| NAND_D3 | GPIO_1[31] | NAND_D3 | X | X |
| NAND_D4 | GPIO_2[6] | NAND_D4 | X | X |
| NAND_D5 | GPIO_2[7] | NAND_D5 | X | X |
| NAND_D6 | GPIO_2[8] | NAND_D6 | X | X |
| NAND_D7 | GPIO_2[9] | NAND_D7 | X | X |
| NAND_REN | GPIO_2[0] | NAND_REN | X | X |
| NAND_CEN0 | GPIO_2[1] | NAND_CEN0 | X | X |
| NAND_CEN1 | GPIO_2[2] | NAND_CEN1 | X | X |
| NAND_CLE | GPIO_2[3] | NAND_CLE | X | X |
| NAND_ALE | GPIO_2[4] | NAND_ALE | X | X |
| NAND_WEN | GPIO_2[5] | NAND_WEN | X | X |

**Table 4-46. Pin-mux Table 9**

| macphy_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| EDP_LINKLED | GPIO_2[14] | EDP_LINKLED | IR-Det[18] | X |
| EDP_SPDLED | GPIO_2[13] | EDP_SPDLED | IR-Det[17] | X |

**Table 4-47. Pin-mux Table 10**

| TV_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| TV_D0 | GPIO_0[25] | MII_RXER | TV_D0 | X |
| TV_D1 | GPIO_0[26] | MII_CRS | TV_D1 | X |
| TV_D2 | GPIO_2[10] | MII_RXDV | TV_D2 | X |
| TV_D3 | GPIO_2[15] | MII_RXD[0] | TV_D3 | X |
| TV_D4 | GPIO_2[16] | MII_RXD[1] | TV_D4 | X |
| TV_D5 | GPIO_2[17] | MII_RXCK | TV_D5 | X |
| TV_D6 | GPIO_2[18] | MII_TXCK | TV_D6 | X |
| TV_D7 | GPIO_2[19] | MII_TXD[0] | TV_D7 | X |
| TV_D8 | GPIO_2[20] | MII_TXD[1] | TV_D8 | X |
| TV_D9 | GPIO_2[21] | MII_TXEN | TV_D9 | X |
| TV_D10 | GPIO_0[22] | MII_COL | TV_D10 | X |
| TV_D11 | GPIO_0[23] | MII_CLKOUT | TV_D11 | X |
| TV_D12 | GPIO_0[24] | MII_MDC | TV_D12 | X |
| TV_D13 | GPIO_0[25] | MII_MDIO | TV_D13 | X |
| TV_D14 | GPIO_0[26] | MII_PHYLINK | TV_D14 | X |
| TV_D15 | GPIO_0[27] | MII_PDNPHY | TV_D15 | X |

**Table 4-48. Pin-mux Table 11**

| mii_mux | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| MII_RXD[2] | GPIO_2[28] | MII_RXD[2] | - | - |
| MII_RXD[3] | GPIO_2[29] | MII_RXD[3] | - | - |
| MII_TXD[2] | GPIO_2[30] | MII_TXD[2] | TV_PCLK | - |
| MII_TXD[3] | GPIO_2[31] | MII_TXD[3] | TV_PCLK | - |

# Chapter 5

# FA626TE

This chapter contains the following sections:


- 5.1 General Description
- 5.2 Block Diagram

## 5.1 General Description

FA626TE is an ultra-high speed, general-purpose, 32-bit embedded RISC processor. It includes a CPU core, an instruction cache, a data cache, an instruction scratchpad, a data scratchpad, a write buffer, a Memory Management Unit (MMU), and a JTAG ICE interface. The CPU core is based on the Harvard architecture design with eight pipeline stages. To improve the overall performance, the FA626TE CPU core also contains a Branch Target Buffer (BTB) and Return Stack (RS) to reduce the branch penalty.

The FA626TE CPU core is instruction-compatible with the ARM V5TE® architecture. It uses four 32-bit wide AHB interfaces to communicate with the external memory and devices. The FA626TE CPU core is suitable for various applications, especially those requiring high performance and low power consumption.

## 5.2 Block Diagram

The block diagram of the FA626TE CPU core is shown in Figure 5-1. The detailed description of each functional block is addressed in the following sections.

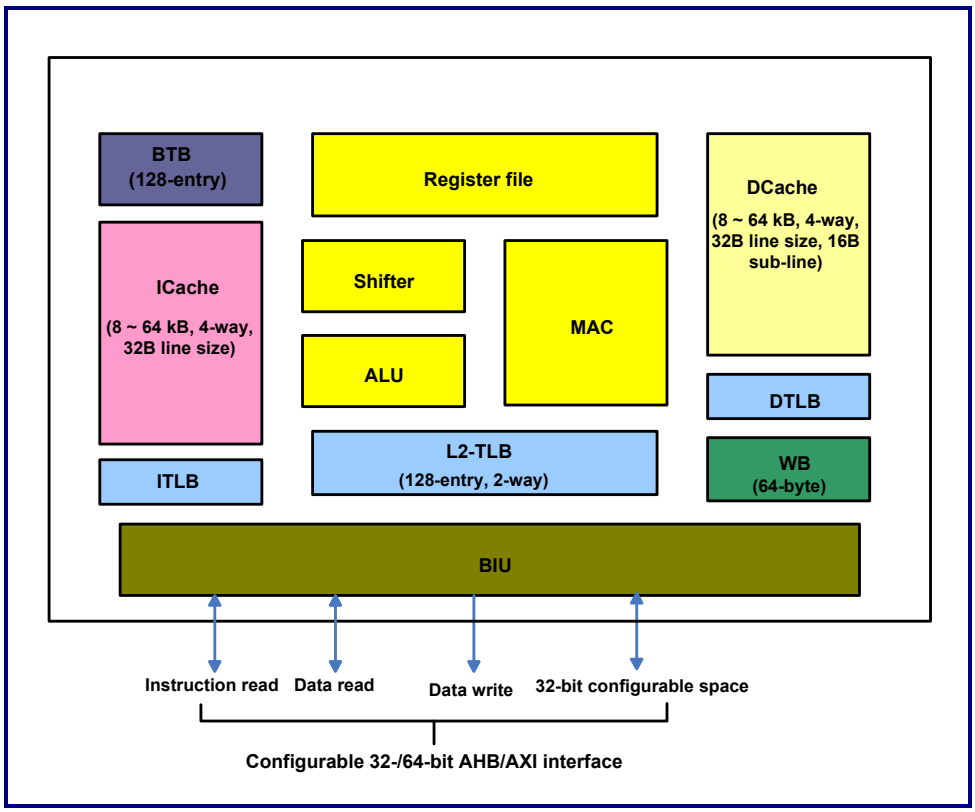



**Figure 5-1. Functional Block Diagram of FA626TE**

### 5.2.1 CPU Core

The FA626TE cores are fully compliant with the ARM V5TE® architecture, which includes the V5TE instruction set. The ARM V5TE® instruction set is well-documented in the ARM Architecture Reference Manual, second edition. The CPU cores adopt the Harvard architecture design with eight-stage pipelines, which include the Fetch, Instruction, Decode, Register Access, Shift, Execution, Memory, and Write stages. There are a total of 30 general-purpose registers and 6 processor status registers. The FA626TE cores provide seven processor modes, which include the Supervisor, System, FIQ, IRQ, Abort, Undefined, and User modes.

### 5.2.2 Branch Prediction Unit (BPU)

The Branch Prediction Unit (BPU) is used to improve the performance of the processor through three branch prediction mechanisms: Branch Target Buffer (BTB), Return Stack (RS), and Static Branch Prediction (SBP). With accurate branch prediction, BTB can resolve most control dependencies and reduce branch penalties. The FA626TE BTB is a 128-entry direct-map structure with 2-bit counter algorithm in the branch prediction that provides all invalidated BTB-entry operations.

RS is a 2-entry buffer that stores the return address of a procedure call. The return address is pushed into RS when a procedure call occurs. When the return instruction is received, the return address is popped from RS. An empty RS gives no prediction.

SBP is used when a branch misses BTB. SBP always predicts when the unconditional branches (B/BL) are taken. For the conditional branches, the backward branches are predicted as taken; and the forward branches are predicted as not taken.

### 5.2.3 Instruction Cache (ICache)

The Instruction Cache (ICache) applies the locality of a program to improve the performance of a processor. The recently used instructions are stored in a faster local memory, ICache. As a result, when the instructions are executed next time, the processor can access the instructions from ICache instead of taking a long access latency to fetch the instructions from a slower external memory. ICache has a 4-way set-associate architecture that can be configured as 4K, 8K, 16K, 32K, or 64K in size. A simplified Least Recently Used (LRU) algorithm will be used as the replacement strategy. The FA626TE cores provide several management instructions for ICache, which include invalidation, pre-fetch, and locking.

### 5.2.4 Data Cache (DCache)

The Data Cache (DCache) applies to the locality of a program to improve the performance of the processor. The recently used data are stored in a faster local memory, DCache. As a result, when data are needed next time, the processor will access data from DCache, instead of accessing them from a slower external memory. DCache has a 4-way set-associate architecture that can be configured as 4K, 8K, 16K, 32K, or 64K in size. A simplified LRU algorithm will be implemented as the replacement strategy. The FA626TE core provides several management instructions for DCache, which include invalidation, clean, pre-fetch, and locking.

### 5.2.5 Memory Management Unit (MMU)

MMU provides the address translation and check permission mechanisms for accessing the memories. The FA626TE MMU has a 2-level TLB structure. Level-1 TLB includes an ITLB for the instruction access look-up and a DTLB for the data access look-up. Both look-ups are 8-entry fully-associate. Level-2 TLB (UTLB) is a unified 2-way set-associate TLB structure. The TLBs cache the recently used page descriptors for the address translation. It greatly improves the overall performance. Once UTLB misses, the page table walk will be completed automatically by the hardware.

The FA626TE MMU is compatible with MMU defined in the ARM Architecture Reference Manual, second edition, with minor differences in the cache and TLB implementation.

### 5.2.6 IScratchpad and DScratchpad

A scratchpad is a fast on-chip SRAM located near the processor core. The performance-critical code or data is pre-fetched to the scratchpads and be executed at full CPU speed. This is particularly useful for the embedded applications. The size of a scratchpad can be 1 kB, 2 kB, 4 kB, 8 kB, 16 kB, 32 kB, 64 kB, or 128 kB. When the first scratchpad instruction is accessed, the FA626TE CPU will automatically fill the entire IScratchpad. DScratchpad needs to be filled by software. The sizes of the IScratchpad and DScratchpad can be programmed as 1 kB, 2 kB, 4 kB, 8 kB, 16 kB, 32 kB, 64 kB, or 128 kB. The base addresses of IScratchpad and DScratchpad can be programmed with the CR9 register by using the physical address. Both IScratchpad and DScratchpad of FA626TE are removable.

### 5.2.7 Bus Interface Unit (BIU)

The Bus Interface Unit (BIU) accepts the requests from the CPU core to access the memories and execute the instructions and data accesses through the external system bus. This unit has four bus ports that are designed to provide a high-bandwidth interface. The Instruction Read (IR) port is used for the instruction reading. The Data Read (DR), Data Write (DW), and Peripheral Ports are used to access data. Each port is an AHB or AXI interface.

### 5.2.8 Write Buffer (WB)

The Write Buffer (WB) hides the write latency to the next level of the memory hierarchy and reduces the write traffic. Without the write buffer, the CPU cores stall when writing to a slower external memory. The write buffer accommodates the slower memory writes so that the CPU cores do not have to wait until the write operation is complete. This improves the overall performance. The FA626TE write buffer includes an address buffer and a data buffer. The size of the data buffer is eight double-words, while the size of the address buffer is eight words.

### 5.2.9 ICE

A FA626TE core provides an ARM-compatible JTAG-style ICE interface. The ICE module debugs the FA626TE program through this interface. The standard ARM debuggers, such as AXD™ or RealView™, can be used for debugging.

### 5.2.10 Power-saving Control Unit

The power-saving control unit controls the processor global clocking to reduce the operating power. It also provides the power-saving mode. When the software program detects that the CPU is in the idle state for a given amount of time, it can force the processor to enter the power-saving mode. In the power-saving mode, the processor stops the instruction execution and enters an idle state. The system logic can then stop the whole processor clock to save power. While in the power-saving mode, the processor can be awoken through either an interrupt or an ICE activity. The start-up time is 16 cycles if the processor is awoken from the idle mode.

### 5.2.11 Coprocessor Transfer Instruction

FA626TE only supports the MCR and MRC instructions for CP15 (MMU), CP14 (ICE), and CP8 (Power-on trap). All other coprocessor-related instructions, such as LDC, SDC, CDP, LDC2, STC2, MCR2, MRC2, MCRR, MRRC, or a coprocessor other than CP8, CP14, or CP15, will cause undefined exceptions.

# Chapter 6

# AHB Controller

This chapter contains the following sections:


- 6.1 General Description
- 6.2 Features
- 6.3 Block Diagram
- 6.4 Memory Map/Register Definition
- 6.5 Functional Description
- 6.6 Multiplexer
- 6.7 Register Slave

## 6.1 General Description

The building blocks of the AHB controller include an arbiter, a decoder, a multiplexer, and a register slave. The role of an arbiter in an Advanced Microcontroller Bus Architecture (AMBA) system is to grant the master the right to access the bus. Each bus master has a REQUEST/GRANT signal to interface with the arbiter, and the arbiter uses a prioritization scheme to decide which AHB master has the highest priority to request a bus. The decoder in the AHB system performs a centralized address decoding function that improves the portability of peripherals by making them independent of the system memory map. The multiplexer is used to route the control signals and write data from the masters to the slaves. It also routes the response signals and reads data from the slaves to the masters. The register slave contains some registers. Users can change the function of the AHB controller by programming these registers.

## 6.2 Features

- Compliant with AMBA 2.0 Specification
- Two-level round-robin priority arbitration
- Configurable number (1 ~ 15) of AHB masters
- Configurable number (1 ~ 32) of AHB slaves
- Configurable 32-bit/64-bit AHB data bus

## 6.3 Block Diagram

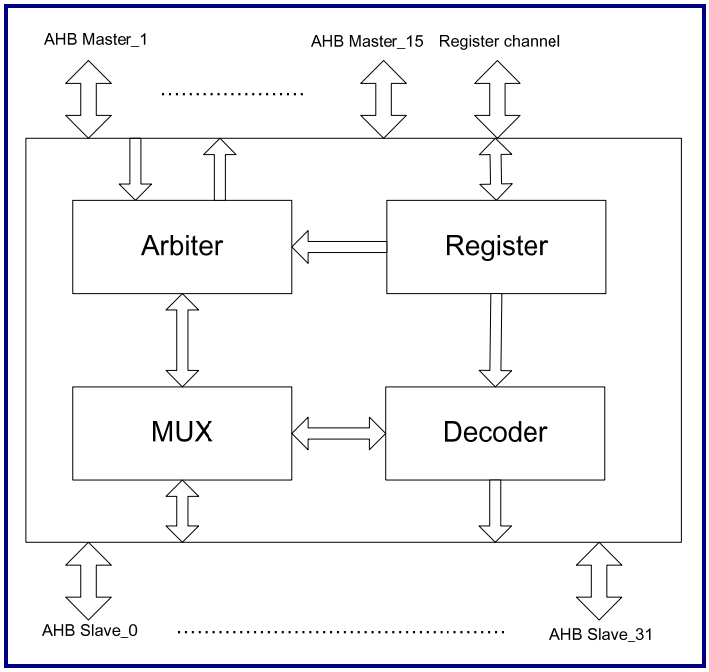



**Figure 6-1. Block Diagram of AHB Controller**

## 6.4 Memory Map/Register Definition

### 6.4.1 Summary of Control Registers

**Table 6-1. Summary of Control Registers**

| Address | I/O Type | Description | Reset Value |
|---|---|---|---|
| + 0 x 4 * *n* | R/W | Base address and memory size register | Depending on the configurations (*n* = 0 ~ 31) |
| + 0 x 80 | R/W | Priority level register | 0x0000_0000 |
| + 0 x 84 | R/W | Idle count register | 0x0000_0001 |
| + 0 x 88 | R/W | Control register | 0x 0010_0F00 |
| + 0 x 8C | R | Revision register | - |
| + 0 x 90 | R/W | AHB bus request enable register | 0x0000_FFFF |

### 6.4.2 AHB Slave *n* Base/Size Register

This register configures the base address/space size of the AHB slave *n*. The base/size registers for all the slaves have the same format and the same definition as the bit fields. Table 6-2 lists the bit assignment of the AHB slave *n* base/size register.

**Table 6-2. AHB Slave *n* Base/Size Register**

| Bit | Name | I/O Type | Description |
|---|---|---|---|
| [31:20] | BaseAddr | R/W | Base address of [31:20] |
| [19:16] | SizeAddr | R/W | Size of address space<br>0000: 1 MB<br>0001: 2 MB<br>0010: 4 MB<br>0011: 8 MB<br>0100: 16 MB<br>0101: 32 MB<br>0110: 64 MB<br>0111: 128 MB<br>1000: 256 MB<br>1001: 512 MB<br>1010: 1 GB<br>1011: 2 GB<br>1001 ~ 1111: Reserved |
| [15:0] | - | - | Reserved |

Notes:

1. The BaseAddr/SizeAddr bits of all the slaves have the same format and the same definition as the bit fields.
2. The reset value of slave *n* is defined by (FTAHBC020S_AHB_SLV*n*_BASE|FTAHBC020S_AHB_SLAVE*n*_SIZE), where FTAHBC020S_AHB_SLV*n*_BASE and FTAHBC020S_AHB_SLAVE*n*_SIZE are the hardware-configured variables.
3. The setting value of the base address must be a multiple of the space size setting value. For example, if the base address of [31:20] is set to 0x002, then the base address is 2M. Therefore, the space size must be 1M or 2M. That is, the space size must be 0 or 1.

### 6.4.3 Priority Control Register

The arbiter supports 2-level priority algorithm to arbitrate the master requests. Each master will be programmed to the high level or the low level. Table 6-3 lists the bit assignments of the priority control register.

**Table 6-3. Priority Control Register**

| Bit | Name | I/O Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:1] | PLevel | R/W | Bit *n* represents the level of master *n*<br>0: Low level, level 0<br>1: High level, level 1 |
| 0 | - | - | Reserved |

### 6.4.4 Idle Count Register

The idle count means the maximum idle command cycles before the re-arbitration when a master is granted. The arbiter will re-arbitrate if a granted master does not issue a NON-SEQ command within the idle count period. Table 6-4 lists the bit assignments of the idle count register.

**Table 6-4. Idle Count Register**

| Bit | Name | I/O Type | Description |
|---|---|---|---|
| [31:6] | - | - | Reserved |
| [5:0] | IdleCount | R/W | This parameter specifies the idle count cycles. |

### 6.4.5 Control Register

Table 6-5 lists the bit assignments of the control register.

**Table 6-5. Control Register**

| Bit | Name | I/O Type | Description |
|---|---|---|---|
| [31:25] | - | - | Reserved |
| 24 | IntrSts | R/W | Interrupt status<br>This bit is set when receiving a non-existing address.<br>0: The interrupt does not occur.<br>1: The interrupt occurs.<br>Writing a '0' to this bit clears the interrupt. |

| Bit | Name | I/O Type | Description |
|---|---|---|---|
| [23:22] | - | - | Reserved |
| [21:20] | Response | R/W | Response status<br>When receiving a non-existing address, the decoder will respond to the master.<br>00: OK response<br>01: ERROR response (Default setting)<br>10: RETRY response<br>11: Not allowed |
| [19:17] | - | - | Reserved |
| 16 | IntsMask | R/W | Interrupt mask<br>Enable or disable the interrupt when the default slave is selected.<br>0: Disable<br>1: Enable |
| [15:8] | INCRLength | R/W | Burst length of INCR<br>When a master issues an INCR command, the arbiter will grant the master to transfer a total amount of (INCR+1) bursts. The default value is '0xF.' Setting to '0' is not allowed.<br>INCR Length Burst Length<br>0x1 0x2<br>0x2 0x3<br>0x3 0x4<br>0x4 0x5<br>: :<br>: :<br>0xff 0x100 |
| [7:1] | - | - | Reserved |
| 0 | Remap | R/W | Remap function<br>Switch the base addresses between slave 4 and slave 6<br>1: Activate remap function<br>After applying the remap function, the new base address of slave 6 = the original base address of slave 4; the new base address of slave 4 = the original base address of slave 4 + the space size of slave 6.<br>Please note that the base address is the boundary of the space size. |

### 6.4.6 AHB Bus Request Enable Register

Table 6-6 lists the bit assignments of the AHB bus request enable register.

**Table 6-6. AHB Bus Request Enable Register**

| Bit | Name | I/O Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:1] | HBusReqEn | R/W | The AHB master bus request signal can be masked by this register. Each bit is for each AHB master correspondingly.<br>HBusReqEn[***n***]:<br>0: The AHB master ***n*** bus request is masked.<br>1: The AHB master ***n*** bus request is not masked. The AHB master ***n*** will be serviced by the AHB controller.<br>Note: HBusReqEn[***n***] is for AHB master ***n***, where ***n*** is 1 ~ 15. |
| 0 | - | - | Reserved |

## 6.5 Functional Description

### 6.5.1 Arbiter

The main function of the arbiter is to arbitrate requests from the AHB masters. If the requests are issued simultaneously, the arbiter uses a prioritized scheme to decide which master currently has the highest priority. The arbiter supports two levels (Level 1 and Level 0) and the round-robin function. Level 1 has higher priority than Level 0. When Level 1 masters and Level 0 masters simultaneously request the same bus, the arbiter will always grant the bus to Level 1 masters first. The round-robin algorithm makes the currently granted master the lowest priority master in the next arbitration on the same level. All masters can be programmed to Level 1 or Level 0.

### 6.5.2 Decoder

The decoder in an AMBA system is used to perform a centralized address decoding function. It decodes the address command of the master and then sends a selection signal to the device being accessed by the master.

## 6.6 Multiplexer

This block serves mainly as a multiplexer that routes the control signals and writes data from the AHB masters to the AHB slaves. It also routes the response signals and reads data from the AHB slaves to the AHB masters.

## 6.7 Register Slave

The register slave includes several control registers of the AHB controller. Users can program these registers for specific system requirements via the AHB register port. HSIZE should be WORD(3'b2) when accessing registers if the system operates in the big-endian mode.

# Chapter 7

# APB Bridge

This chapter contains the following sections:


- 7.1 General Description
- 7.2 Features
- 7.3 Block Diagram
- 7.4 DMA Routing Table
- 7.5 Memory Map/Register Definition
- 7.6 Functional Description

## 7.1 General Description

The main purpose of the APB Bridge is to convert the transactions between the APB and AHB buses. The AHB interface of this APB Bridge is fully compliant with the AMBA 2 AHB protocol; while the APB interface of this APB Bridge is fully compliant with the AMBA 3 APB protocol. The APB Bridge is the only bus master on the APB bus and it serves as a slave on the AHB bus. This APB Bridge can support a maximum of 32 configurable APB devices. In addition, it provides the DMA function that by using the hardware configurations. 15 sets of the DMA requests/grants and four configurable DMA channels are available to the end users. The APB Bridge contains a set of control registers, two AHB slave interfaces (One for accessing the control registers and the other for accessing the APB devices), an AHB master interface for DMA functions, a DMA engine, and an arbiter.

## 7.2 Features

- Compliant with AMBA 2 AHB protocol for AHB interface
- Compliant with AMBA 3 APB protocol for APB interface
- Backward compatible with AMBA 2 APB protocol
- Supports up to 32 APB devices that can be programmed by using hardware configuration
- Hardware configurable DMA functions
- Supports the synchronous mode between AHB and APB interfaces

## 7.3 Block Diagram

Figure 7-1 shows the functional block diagram of the APB Bridge.

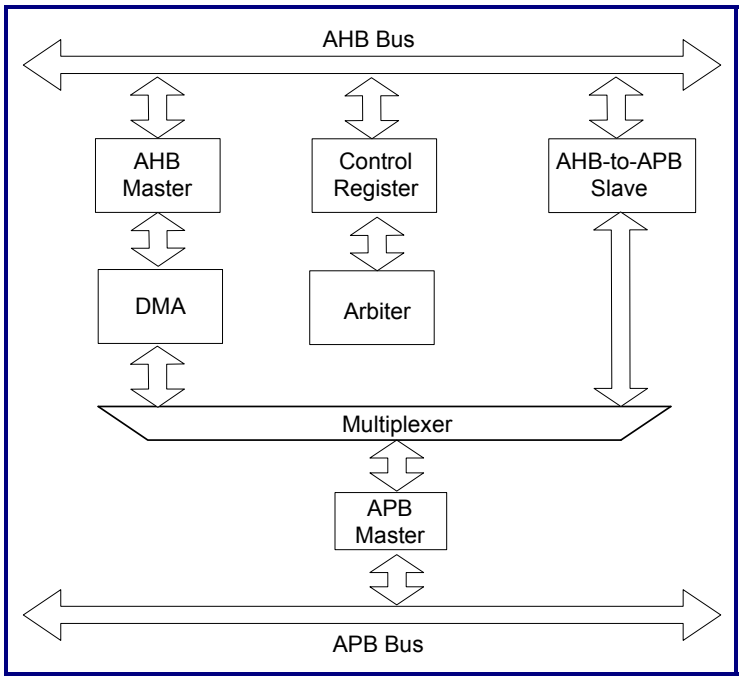



**Figure 7-1. Functional Block Diagram of APB Bridge**

## 7.4 DMA Routing Table

DMA provides 16 req/ack signal pairs. Every AHB device with the DMA access capability is associated with a specific req/ack pair. The following table lists the relationship of the AHB devices and req/ack pairs.

**Table 7-1. APB DMA REQ/ACK Mapping Table**

| APB DMA Request No. | APB Device |
|---|---|
| 1 | UART_0 RX |
| 2 | UART_0 TX |
| 3 | UART_1 RX |
| 4 | UART_1 TX |
| 5 | UART_2 RX |
| 6 | UART_2 TX |
| 7 | UART_3 RX |

| APB DMA Request No. | APB Device |
|---|---|
| 8 | UART_3 TX |
| 9 | UART_4 RX |
| 10 | UART_4 TX |
| 11 | Reserved |
| 12 | PWMTMR1 |
| 13 | PWMTMR2 |
| 14 | PWMTMR3 |
| 15 | PWMTMR4 |

## 7.5 Memory Map/Register Definition

Table 7-2 summarizes the APB Bridge registers.

**Table 7-2. Summary of APB Bridge Registers**

| Address | Type | Description | Reset Value |
|---|---|---|---|
| 4 * *n* | R/W | APB slave *n* base/size register | Depend on the configuration (*n* = 0 ~ 31) |
| + 0x80 | R/W | Source address of the DMA channel A | 0 x 0000_0000 |
| + 0x84 | R/W | Destination address of the DMA channel A | 0 x 0000_0000 |
| + 0x88 | R/W | Cycles of the DMA channel A | 0 x 0000_0000 |
| + 0x8C | R/W | Command of the DMA channel A | 0 x 0000_0000 |
| + 0x90 | R/W | Source address of the DMA channel B | 0 x 0000_0000 |
| + 0x94 | R/W | Destination address of the DMA channel B | 0 x 0000_0000 |
| + 0x98 | R/W | Cycles of the DMA channel B | 0 x 0000_0000 |
| + 0x9C | R/W | Command of the DMA channel B | 0 x 0000_0000 |
| + 0xA0 | R/W | Source address of the DMA channel C | 0 x 0000_0000 |
| + 0xA4 | R/W | Destination address of the DMA channel C | 0 x 0000_0000 |
| + 0xA8 | R/W | Cycles of the DMA channel C | 0 x 0000_0000 |
| + 0xAC | R/W | Command of the DMA channel C | 0 x 0000_0000 |
| + 0xB0 | R/W | Source address of the DMA channel D | 0 x 0000_0000 |
| + 0xB4 | R/W | Destination address of the DMA channel D | 0 x 0000_0000 |
| + 0xB8 | R/W | Cycles of the DMA channel D | 0 x 0000_0000 |
| + 0xBC | R/W | Command of the DMA channel D | 0 x 0000_0000 |
| + 0xC0 | R/W | APB control register | 0 x 0000_0000 |
| + 0xC4 | R/WOC | APB status register | 0 x 0000_0000 |
| + 0xC8 | RO | Revision register | - |

The following sections describe the APB Bridge registers in detail.

### 7.5.1 Base/Size Register of APB Slave *n*

This register configures the base address/space size (*n* = 0 ~ 31) of the APB slave *n*. The values of the base/size register of all the slaves have the same format and same definition as the fields. Table 7-3 lists the bit assignments.

**Table 7-3. Base/Size Register of APB Slave *n***

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:20] | Base | R/W | Base addresses[29:20] |
| [19:16] | Size | R/W | Size of the address space<br>0000: 1M<br>0001: 2M<br>0010: 4M<br>0011: 8M<br>0100: 16M<br>0101: 32M<br>0110: 64M<br>0111: 128M<br>1000: 256M<br>1001 ~ 1111: Reserved |
| [15:0] | - | - | Reserved |

### 7.5.2 Source Addresses of DMA Channels A/B/C/D (Optional for DMA)

This register configures the source addresses of the DMA channels A/B/C/D. Table 7-4 lists the bit assignments.

**Table 7-4. Source Addresses of DMA Channel Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | *n*SrcAdr | R/W | Source addresses of the current DMA cycles |

### 7.5.3 Destination Addresses of DMA Channels A/B/C/D (Optional for DMA)

This register configures the destination addresses for the DMA channels A/B/C/D. Table 7-5 lists the bit assignments.

**Table 7-5. Destination Addresses of DMA Channel Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | *n*DesAdr | R/W | Destination addresses of the current DMA cycles |

### 7.5.4 Cycle of DMA Channels A/B/C/D (Optional for DMA)

This register configures the cycles of DMA channels A/B/C/D. Table 7-6 lists the bit assignment.

**Table 7-6. Cycles of DMA Channels Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | - | - | Reserved |
| [23:0] | *n*Cyc | R/W | A DMA cycle consists of one or four bus data transfer cycles, depending on the DMA command burst bit (*n*BurMod) in the DMA channel command register. |

### 7.5.5 Command of DMA Channels A/B/C/D (Optional for DMA)

This register configures the commands of DMA channels A/B/C/D. Table 7-7 lists the bit assignment.

**Table 7-7. Commands of DMA Channels Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:28] | - | - | Reserved |
| [27:24] | *n*SReqsel | R/W | Request/Grant signal for selecting the source addresses of the DMA hardware handshake mode<br>0: No request/grant signal<br>1: Select ps1_preq/ps1_pgnt as the source of DMA hardware handshake signals<br>2: Select ps2_preq/ps2_pgnt as the source of DMA hardware handshake signals<br>:<br>15: Select ps15_preq/ps15_pgnt as the source of DMA hardware handshake signals |
| [23:22] | - | - | Reserved |
| [21:20] | *n*DataWidth | R/W | Data width of the transfer (When the DMA source or the destination is the APB device, the data width must be set to word. The APB device supports the word-width transfer only.)<br>00: Word<br>01: Half-word<br>10: Byte<br>11: Reserved |
| [19:16] | *n*DReqSel | R/W | The request/grant signal selects the destination address in the DMA hardware handshake mode.<br>0: No request/grant signal<br>1: Select ps1_preq/ps1_pgnt as the destination of the DMA hardware handshake signals<br>2: Select ps2_preq/ps2_pgnt as the destination of the DMA hardware handshake signals<br>:<br>15: Select ps15_preq/ps15_pgnt as the destination of the DMA hardware handshake signals |
| 15 | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [14:12] | *n*DesAdrInc | R/W | Destination address incremental<br>000: No increment<br>001: +1 (Burst = 0), +4 (Burst = 1)<br>010: +2 (Burst = 0), +8 (Burst = 1)<br>011: +4 (Burst = 0), +16 (Burst = 1)<br>101: -1 (Burst = 0), not supportted in the burst mode (Burst = 1)<br>110: -2 (Burst = 0), not supportted in the burst mode (Burst = 1)<br>111: -4 (Burst = 0), not supportted in the burst mode (Burst = 1)<br>Note: When DMA is configured in the decreasing mode, the destination address cannot cross the 1K boundary within one DMA transfer period. |
| 11 | - | - | Reserved |
| [10:8] | *n*SrcAdr | R/W | Source address incremental<br>000: No increment<br>001: +1 (Burst = 0), +4 (Burst = 1)<br>010: +2 (Burst = 0), +8 (Burst = 1)<br>011: +4 (Burst = 0), +16 (Burst = 1)<br>101: -1 (Burst = 0), not supportted in the burst mode (Burst = 1)<br>110: -2 (Burst = 0), not supportted in the burst mode (Burst = 1)<br>111: -4 (Burst = 0), not supportted in the burst mode (Burst = 1)<br>Note: When DMA is configured in the decreasing mode, the source address cannot cross the 1K boundary within one DMA transfer period. |
| 7 | *n*DesAdrSel | R/W | Destination address control<br>0: APB<br>1: AHB |
| 6 | *n*SrcAdrSel | R/W | Source address control<br>0: APB<br>1: AHB |
| 5 | *n*ErrIntEnb | R/W | Interrupt when the AHB slave returns an error response during a DMA transfer<br>0: Disable<br>1: Enable |
| 4 | *n*ErrIntSts | R/W | Error response interrupt flag<br>0: No interrupt occurs<br>1: Interrupt occurs<br>This bit can be manually cleared by writing a ‘0’ to it. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 3 | *n*BurMod | R/W | Burst mode control<br>0: No burst; each DMA cycle consists of one single-bus transfer cycle.<br>1: Burst; each DMA cycle consists of four bus transfer cycles. |
| 2 | *n*FinIntEnb | R/W | Finished interrupt control<br>0: Disable<br>1: Enable |
| 1 | *n*FinIntSts | R/W | Finishing interrupt flag<br>0: No interrupt occurs<br>1: Interrupt occurs<br>This bit can be manually cleared by writing '0' to it. |
| 0 | *n*EnbDis | R/W | Enable/Stop DMA transfer<br>This bit can be automatically cleared when all DMA cycles are finished.<br>0: Stop the DMA channel<br>1: Enable the DMA channel<br>Note: When CPU stops the DMA transfers by setting this bit to '0' before all DMA cycles are finished. Before CPU reconfigures another DMA transfer, CPU needs to monitor this bit to ensure that this bit has been cleared by the DMA state machine. |

### 7.5.6 APB Control Register

**Table 7-8. APB Control Register (0xc0)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:3] | - | - | Reserved |
| 2 | BufWrErrIntEn | R/W | Interrupt enable of bufferable write<br>1: When the APB Bridge receives an error response of an AHB bufferable write command from an APB device, the APB Bridge will assert the interrupt signal, rshint.<br>0: When the APB Bridge receives an error response of an AHB bufferable write command from an APB device, the APB Bridge will not assert the interrupt signal, rshint. |
| [1:0] | HProtType | R/W | APB Bridge write buffer control<br>2'b00: The AHB write data is saved into the write buffer if the attribute of the AHB input port, hprot[1:0], is bufferable.<br>2'b01: The AHB write data is always saved into the write buffer no matter what the attribute of an AHB command, hprot[1:0], will be.<br>2'b10: The AHB write data is always not saved into the write buffer no matter what the attribute of an AHB command, hprot[1:0], will be. |

**Table 7-9. APB Status Register (0xc4)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:1] | - | - | Reserved |
| 0 | BufWrErrSts | R/WOC | Error status flag for an AHB bufferable write command<br>This flag is set when the APB Bridge receives an error of an AHB bufferable write command response from an APB device.<br>This flag is cleared by CPU by writing a '1' to this register. (Write '1' to clear) |

### 7.5.7 Revision Register

This register records the IP version.

**Table 7-10. Revision Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | - | - | Reserved |
| [23:0] | revision | R | IP revision<br>For example, the value of version 1.5.3 is identified as 0x010503. |

## 7.6 Functional Description

The APB Bridge contains a set of control registers, two AHB slave interfaces (One for accessing the control registers and the other for accessing the APB devices), one AHB master interface for DMA functions, one DMA engine, and one arbiter. The following sections describe the functions of each block.

### 7.6.1 Control Register

This block controls the APB Bridge. The APB Bridge can support up to 32 APB devices by the hardware configurations. The base address and the space size of all devices can be set by programming this block. Users can also program the control register to support DMA by the APB Bridge.

### 7.6.2 AHB Slave for Accessing APB Devices

The AHB slave interface converts the AHB-bus transfers into the APB-bus transfers.

### 7.6.3 AHB Slave for Accessing APB Bridge Registers

The AHB slave interface accesses the control registers of the APB Bridge. Please note that the accesses to the control registers can be performed simultaneously with the accesses to the APB devices when two independent AHB buses are adopted.

### 7.6.4 AHB Master

The AHB master performs the AHB bus transfers if DMA of the APB Bridge requests the data transfers of the AHB bus.

### 7.6.5 APB Master

The APB master is the only master on the APB bus that performs the APB transfers. The APB master supports the AMBA 3 APB protocol and is backward compatible to the AMBA 2 APB protocol.

### 7.6.6 DMA Engine

The APB Bridge supports the DMA functions for the AHB-to-AHB, AHB-to-APB, APB-to-AHB, and APB-to-APB transactions. The DMA engine can support up to four channels and 15 request/grant pins. A channel granted by the arbiter block is the only channel starts transfers. Each channel can be programmed to one of the 15 sets of request/grant pins in the hardware handshake mode to act as the source device or act as the destination device. The main function of the hardware handshake mode is to provide an indication of the device status. When the device is ready to transfer data, the ps*n*_preq signal is set. Then the DMA will reply with a ps*n*_pgnt signal. Users can also disable the hardware handshake mode by programming the register when a DMA transfer is not necessary of referring to the request/grant pins. The standard APB 2.0 does not support the transfers with non-WORD size. However, the DMA engine of FTAPBBRG020S does support that. For BYTE transfers, the engine puts data of the target byte in the least significant byte of data bus (pwdata[7:0] or psn_prdata[7:0]). For HALFWORD transfers, the engine puts data of the target halfword in the least significant halfword of data bus. (pwdata[15:0] or psn_prdata[15:0]).

### 7.6.7 DMA Hardware Handshake Mode

In this mode, the handshake protocol is followed between DMA and other devices to complete the DMA transfer. Both the source and the destination devices can be programmed in the hardware handshake mode when users set the *n*SReqsel/*n*DReqsel bits in the command register to individually select the source and destination request signals, ps*n*_preq (*n* = 1 ~ 15) for reference. After the *n*EnbDis bit is set, DMA will start to transfer when the request signal (ps*n*_preq) is active from both the source and the destination devices. After one DMA cycle transfer is completed, DMA will assert the grant signal (ps*n*_pgnt) for the source and the destination devices. When the device detects the grant signal, it will de-assert its request signal. After detecting the de-assertion of the request signal, DMA will de-assert the grant signal. Each handshake communication can cause each DMA cycle transfer. After the total count of the DMA cycles is completed, it will assert the interrupt. Figure 7-2 shows the timing diagram of the hardware handshake protocol.

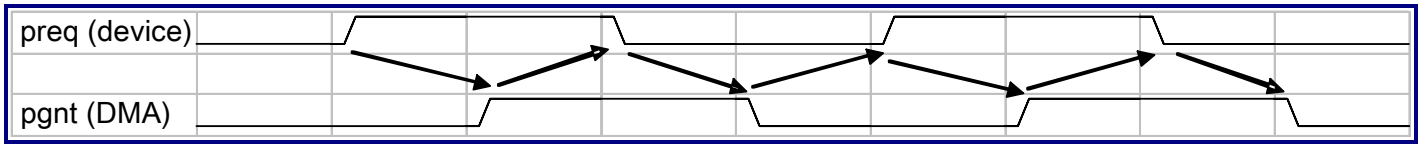



**Figure 7-2. Hardware Handshake Protocol**

### 7.6.8 Arbiter

The arbitration mechanism is used to ensure that only one channel has the right to access the bus at one time. The arbiter performs this function by observing a number of different requests to use the bus and deciding which channel will have the highest priority. The round robin algorithm is implemented to arbitrate the requests of all the DMA channels. That is, the DMA channel, of which the request is currently being granted, will have the lowest priority in the next arbitration.

# Chapter 8

# Static Memory Controller

This chapter contains the following sections:


- 8.1 Features
- 8.2 Block Diagram
- 8.3 Connection of Different Device Types
- 8.4 Programming Model

The static memory controller provides the interface to access the ROM, NOR type Flash, and asynchronous SRAMs.

## 8.1 Features

- Supports the zero-wait-state write function
- Shadows the first bank with other banks
- Wide address range up to 256M bytes
- Supports the ROM, NOR type flash, burst-ROM, asynchronous SRAM, and VLIO devices
- Supports big-endian/little-endian

## 8.2 Block Diagram

Figure 8-1 shows the functional block diagram of the Faraday static memory controller.

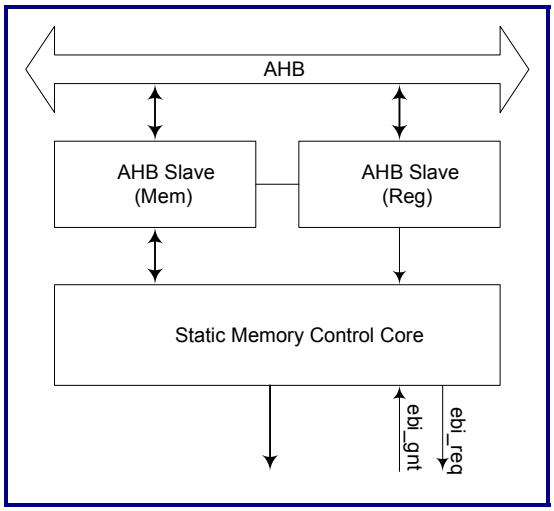



**Figure 8-1. Functional Block Diagram**

The main building blocks of the static memory controller are the AHB slave ports (Memory and register) and the controller engine. The following sections contain the detailed descriptions of each building block.

### 8.2.1 AHB Slave

This block acts as an interface between AHB and the controller engine. It consists of one AHB memory slave to access the memory and one AHB register slave to access the registers of the static memory controller. The following descriptions focus on the memory slave.

In the read mode, the command from AHB will be decoded and the memory read command will be issued by this block to the static memory control engine. Data from the static memory control engine will be latched into FIFO and pushed to AHB.

In the write mode, data and addresses will be latched into FIFO and HREADY will not be pulled low until the internal FIFO is full. Consecutive write operations will be completed without any wait cycles (Zero-wait-state write). The AHB slave will write data to the static memory later.

### 8.2.2 Control Engine

The control engine is used to generate the static memory access cycles, which include the memory read and write operations. The timing parameter is used to control the access cycles to meet the specification of the access timing of the static memory. The data read from the static memory will be latched into the control engine and returned to the AHB slaves.

To share the static memory address/data pins with other IPs, the control engine has to control the external bus req/ack to request the usage of the address and the data bus.

## 8.3 Connection of Different Device Types

The static memory controller supports the asynchronous devices. These devices have different types of connections. The following sections provide an overview on the connection scheme.

### 8.3.1 Asynchronous Devices

Figure 8-2 shows the connection scheme of the asynchronous devices.

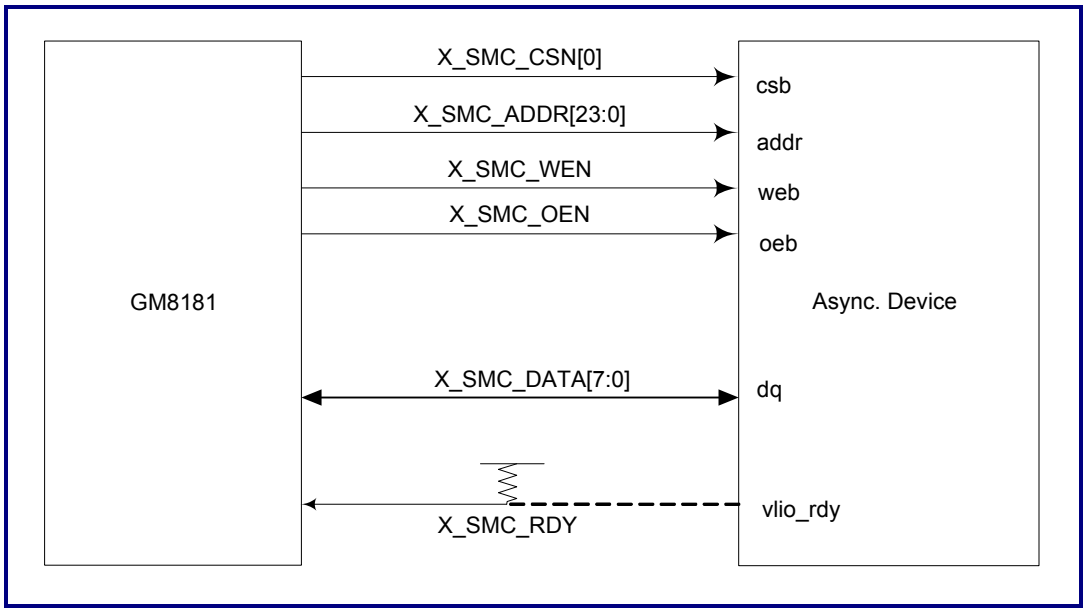



**Figure 8-2. Connection Scheme of Asynchronous Devices**

## 8.4 Programming Model

**Table 8-1. Summary of Static Memory Controller Register**

| Address Offset | Type | Description |
|---|---|---|
| 0x00 | R/W | Configuration register of memory bank 0 |
| 0x04 | R/W | Timing parameter register of memory bank 0 |
| 0x08 | R/W | Configuration register of memory bank 1 |
| 0x0C | R/W | Timing parameter register of memory bank 1 |
| 0x10 | R/W | Configuration register of memory bank 2 |
| 0x14 | R/W | Timing parameter register of memory bank 2 |
| 0x18 | R/W | Configuration register of memory bank 3 |
| 0x1C | R/W | Timing parameter register of memory bank 3 |
| 0x20 | R/W | Configuration register of memory bank 4 |
| 0x24 | R/W | Timing parameter register of memory bank 4 |
| 0x28 | R/W | Configuration register of memory bank 5 |

| Address Offset | Type | Description |
|---|---|---|
| 0x2C | R/W | Timing parameter register of memory bank 5 |
| 0x30 | R/W | Configuration register of memory bank 6 |
| 0x34 | R/W | Timing parameter register of memory bank 6 |
| 0x38 | R/W | Configuration register of memory bank 7 |
| 0x3C | R/W | Timing parameter register of memory bank 7 |
| 0x40 | R/W | Shadow status register |
| 0x80 | R | Revision register |
| 0x84 | R | Feature register |

### 8.4.1 Configuration Register of Memory Bank *n* (Offset = n * 8)

This register contains the base, memory type, memory bus width, and size of the corresponding external memory bank. Table 8-2 shows the bit assignments of the bank configuration register.

**Table 8-2. Memory Bank Configuration Register (Offset = n * 8)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | - | - | Reserved |
| 28 | BNK_EN | R/W | Bank enable flag<br>1: Enabled<br>0: Disabled |
| [27:15] | BNK_BASE | R/W | Bank base address<br>The value of this field equals to the address bits[27:15] on AHB. |
| [14:12] | - | - | Reserved<br>Writing to this field takes no effect. Reading from this field will return a value of zero. |
| 11 | BNK_WPROT | R/W | Bank write protected<br>If BNK_WPROT is set as '1', the corresponding memory bank cannot be written. Any write operation to the protected bank will cause an ERROR response on AHB.<br>If BNK_WPROT is set as '0', the corresponding bank can be read or written. |
| 10 | BNK_TYP1 | R/W | Bank type 1<br>This bit indicates that the devices on the corresponding bank are either synchronous or asynchronous.<br>0: Asynchronous<br>1: Synchronous |

| Bit | Name | Type | Description |
|---|---|---|---|
| 9 | BNK_TYP2 | R/W | Bank type 2<br>If BNK_TYP1 is set to ‘0’, this bit indicates that the devices on the corresponding bank are either the general asynchronous devices or the burst ROMs. Setting this bit to ‘1’ indicates the burst ROMs; setting this bit to ‘0’ indicates the general asynchronous devices.<br>If BNK_TYP1 is set to ‘1’, this bit indicates that the devices on the corresponding bank are either pipelined or non-pipelined. Setting this bit to ‘1’ indicates the pipelined devices; setting this bit to ‘0’ indicates the non-pipelined devices. |
| 8 | BNK_TYP3 | R/W | Bank type 3<br>This bit is only valid when BNK_TYP1 is set to ‘1’ and the write latency exists on the corresponding devices (Late-write).<br>If this bit is set to ‘1’ and late-write is enabled; AT1 will decide the depth of late-write.<br>If this bit is set to ‘0’, the late-write will be disabled. |
| [7:4] | BNK_SIZE | R/W | Bank size<br>The following encoding shows the bank size. The bank size other than the following values may cause unexpected errors.<br>1011: 32 kB<br>1100: 64 kB<br>1101: 128 kB<br>1110: 256 kB<br>1111: 512 kB<br>0000: 1 MB<br>0001: 2 MB<br>0010: 4 MB<br>0011: 8 MB<br>0100: 16 MB<br>0101: 32 MB<br>0110: 64 MB |
| [3:2] | - | - | Reserved |
| [1:0] | BNK_MBW | R/W | Memory bus width<br>This field indicates the bus size of the external memory bus. Writing a value of larger than three will be modulo by four.<br>00: Memory data width is 8.<br>01: Memory data width is 16.<br>10: Memory data width is 32.<br>11: Reserved<br>Note: GM8126 only supports the memory data width of 8 bits. |

### 8.4.2 Timing Parameter Register of Memory Bank (Offset = (*n* * 8) + 4)

This register configures the access timing of the corresponding external memory bank. The turn-around cycles specified by users are inserted by specifying $t_{TRNA}$ in the bank timing register. Table 8-3 shows the bit assignments of the timing control register. The default value of each bit depends on the configuration.

**Table 8-3. Memory Bank Configuration Register (Offset = (*n* * 8) + 4)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:28] | ETRNA | R/W | Extend turn-around time<br>The total turn-around time is the cascade of register and the TRNA field. |
| [27:24] | EAT1 | R/W | Extend access time 1<br>The total access time that the static memory controller used is the cascade of register and AT1. |
| [23:21] | - | - | Reserved |
| 20 | RBE | R/W | Read byte-enable<br>If this bit is set to '1,' the byte-enable will be pulled low when read. Otherwise, the byte-enable will be pulled low only for the write operations. |
| [19:18] | AST | R/W | Address setup time<br>This field specifies the necessary latency to assert chip-enable after the assertion of address. |
| [17:16] | CTW | R/W | Write-enable delay after chip-select<br>This field specifies the necessary latency to assert write-enable after the assertion of chip-select. |
| [15:12] | AT1 | R/W | Access time 1<br>This field will specify the latency to latch data (Read) or change data (Write) after the assertion of write-enable when the general asynchronous device is specified. The value must be larger than zero. Setting this field to zero is acceptable but the behavior will be unpredictable.<br>If the device is specified as burst ROM, this field indicates the read latency of the first data.<br>If BNK_TYPE1 is set to '1' (Synchronous devices), this field indicates the depth of late-write and the maximum value is two (Values exceeding two will be reset to zero). |
| [11:10] | - | - | Reserved<br>Writing data to this field has no effect, and zero will be returned when read. |
| [9:8] | AT2 | R/W | Access time 2<br>This field is used to specify the necessary latency to latch the burst read data. This field is available only when the device type is specified as burst ROM. |
| [7:6] | WTC | R/W | Write-enable to chip-select delay<br>This field specifies the necessary latency to de-assert the chip-enable after the de-assertion of write-enable. |

| Bit | Name | Type | Description |
|---|---|---|---|
| [5:4] | AHT | R/W | Address hold time<br>This field specifies the necessary latency to de-assert the address after the de-assertion of chip-select. |
| [3:0] | TRNA | R/W | Turn-around time<br>This field specifies the necessary latency to re-drive the data bus. |

### 8.4.3 Asynchronous Devices Timing Waveform

Figure 8-3 shows the write timing of the asynchronous devices. Table 8-4 through Table 8-9 show the mapping relationships of the register settings and the actually written timings.

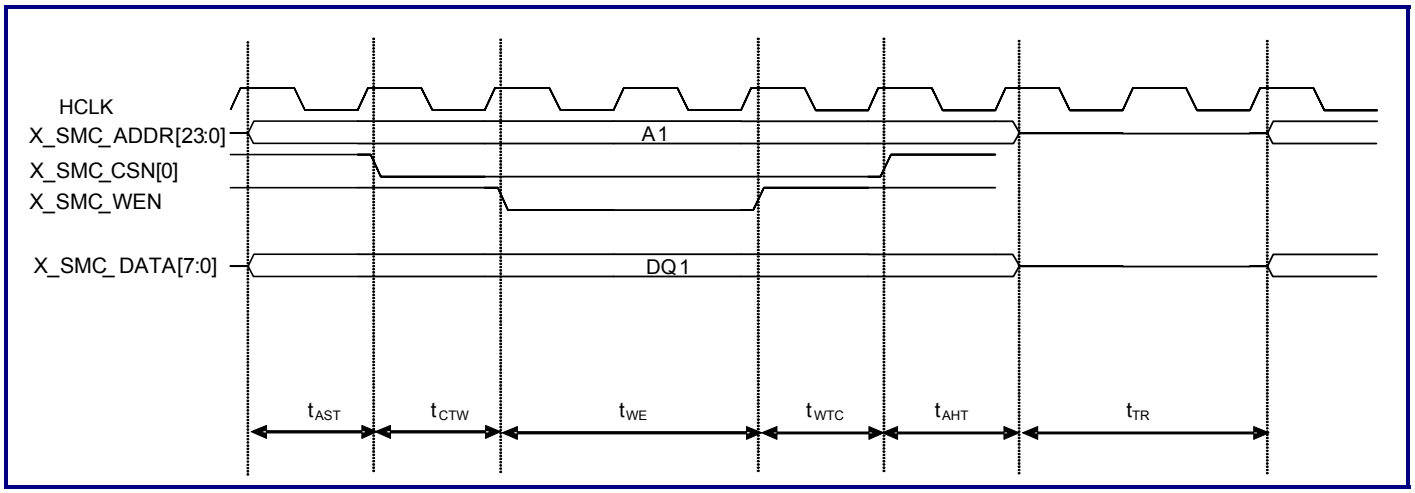



**Figure 8-3. Timing Diagram of SMC Write to Asynchronous Devices**

**Table 8-4. AST Register Setting Mapping**

| Address Setup Time | | | | |
|---|---|---|---|---|
| AST register value | 0 | 1 | 2 | 3 |
| $t_{AST}$ (Unit: hclk) | 0 | 2 | 3 | 4 |

**Table 8-5. CTW Register Setting Mapping**

| Write-enable Delay after Chip–select | | | | |
|---|---|---|---|---|
| CTW register value | 0 | 1 | 2 | 3 |
| $t_{CTW}$ (Unit: hclk) | 0 | 2 | 3 | 4 |

**Table 8-6. AT1 Register Setting Mapping**

| Write Enable Width | | | | | |
|---|---|---|---|---|---|
| AT1 register value | 0 | 1 | 2 | 3 | n |
| $t_{AT1}$ ($t_{WTC}$ and $t_{AHT}$ = 0) (Unit: hclk) | Not allowed | 2 | 3 | 4 | n + 1 |
| $t_{AT1}$ ($t_{WTC}$ or $t_{AHT}$ > 0) (Unit: hclk) | Not allowed | 3 | 4 | 5 | n + 2 |

Note: $t_{WE}$ (Unit: hclk) = 16 * EAT1 + $t_{AT1}$

**Table 8-7. WTC Register Setting Mapping**

| Chip-select Delay after Write-enable | | | | |
|---|---|---|---|---|
| WTC register vlaue | 0 | 1 | 2 | 3 |
| $t_{WTC}$ ($t_{AHT}$ = 0; $t_{TRNA}$ = 0) | 0 | 1 | 2 | 3 |
| $t_{WTC}$ ($t_{AHT}$ = 0; $t_{TRNA}$ > 0) | 0 | 1 | 2 | 3 |
| $t_{WTC}$ ($t_{AHT}$ > 0; $t_{TRNA}$ = 0) | 0 | 2 | 3 | 4 |
| $t_{WTC}$ ($t_{AHT}$ > 0; $t_{TRNA}$ > 0) | 0 | 2 | 3 | 4 |

**Table 8-8. AHT Register Setting Mapping**

| Address Hold Time | | | | |
|---|---|---|---|---|
| AHT register value | 0 | 1 | 2 | 3 |
| $t_{AHT}$ (Unit: hclk) | 0 | 1 | 2 | 3 |

**Table 8-9. TRNA Register Setting Mapping**

| Turn Around Time | | | | | |
|---|---|---|---|---|---|
| TRNA register value | 0 | 1 | 2 | 3 | n |
| $t_{RNA}$ (Unit: hclk) | 2 | 3 | 4 | 5 | n + 2 |

Note: $t_{TR}$ (Unit: hclk) = 16 * ETRNA + $t_{TRNA}$

Figure 8-4 shows the read timing of the asynchronous devices. Table 8-10 through Table 8-15 shows the mapping relationships of the register settings and the actually read timings.

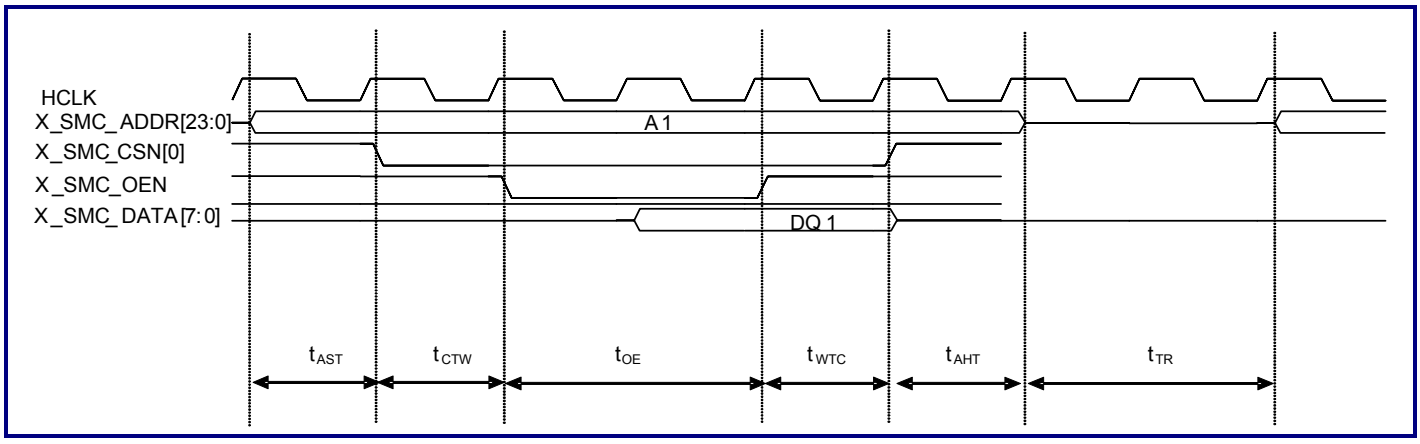



**Figure 8-4. Timing Diagram of SMC Read from Asynchronous Devices**

**Table 8-10. AST Register Setting Mapping**

| **Address Setup Time** | | | | |
|---|---|---|---|---|
| AST register value | 0 | 1 | 2 | 3 |
| $t_{AST}$ (Unit: hclk) | 0 | 2 | 3 | 4 |

**Table 8-11. CTW Register Setting Mapping**

| **Output-enable Delay after Chip-select** | | | | |
|---|---|---|---|---|
| CTW register value | 0 | 1 | 2 | 3 |
| $t_{CTW}$ (Unit: hclk) | 0 | 2 | 3 | 4 |

**Table 8-12. AT1 Register Setting Mapping**

| **Output Enable Width** | | | | | |
|---|---|---|---|---|---|
| AT1 register value | 0 | 1 | 2 | 3 | n |
| $t_{AT1}$ (Unit: hclk) | Not allowed | 2 | 3 | 4 | n + 1 |

Note: $t_{OE}$ (Unit:hclk) = 16 * EAT1 + $t_{AT1}$

**Table 8-13. WTC Register Setting Mapping**

| Chip-select Delay after Read-enable | | | | |
|---|---|---|---|---|
| WTC Register | 0 | 1 | 2 | 3 |
| $t_{WTC}$ ($t_{AHT}$ = 0; $t_{TRNA}$ = 0) | 1 | 3 | 4 | 5 |
| $t_{WTC}$ ($t_{AHT}$ = 0; $t_{TRNA}$ > 0) | 0 | 2 | 3 | 4 |
| $t_{WTC}$ ($t_{AHT}$ > 0; $t_{TRNA}$ = 0) | 1 | 3 | 4 | 5 |
| $t_{WTC}$ ($t_{AHTt}$ > 0; $t_{TRNA}$ > 0) | 1 | 3 | 4 | 5 |

**Table 8-14. AHT Register Setting Mapping**

| Address Hold Time | | | | |
|---|---|---|---|---|
| AHT register value | 0 | 1 | 2 | 3 |
| $t_{AHT}$ (Unit: hclk) | 0 | 1 | 2 | 3 |

**Table 8-15. TRNA Register Setting Mapping**

| Turn Around Time | | | | | |
|---|---|---|---|---|---|
| TRNA register value | 0 | 1 | 2 | 3 | n |
| $t_{TRNA}$ (Unit: hclk) | 2 | 3 | 4 | 5 | n + 2 |

Notes:

1. (ETRNA and TRNA = 0) $t_{TR}$ (Unit: hclk) = 1
2. (ETRNA or TRNA > 0) $t_{TR}$ (Unit: hclk) = 16 * ETRNA + $t_{TRNA}$

### 8.4.4 Timing Waveform of Variable Latency I/O (VLIO) Devices

The VLIO devices are identical to the asynchronous devices except that the smc_vlio_rdy signal is not always tied to logic high. The smc_vlio_rdy signal is connected to the ready pins of VLIO devices. Please note that if the synchronization circuit is applied to the input signal, smc_vlio_rdy, the value of register AT1 shall be set to more than two.

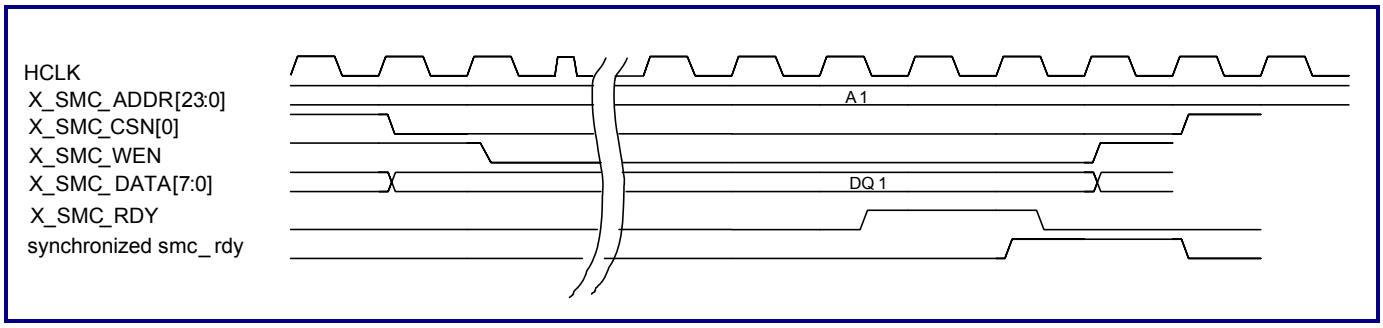



**Figure 8-5. Timing Diagram of SMC Write to VLIO Devices**

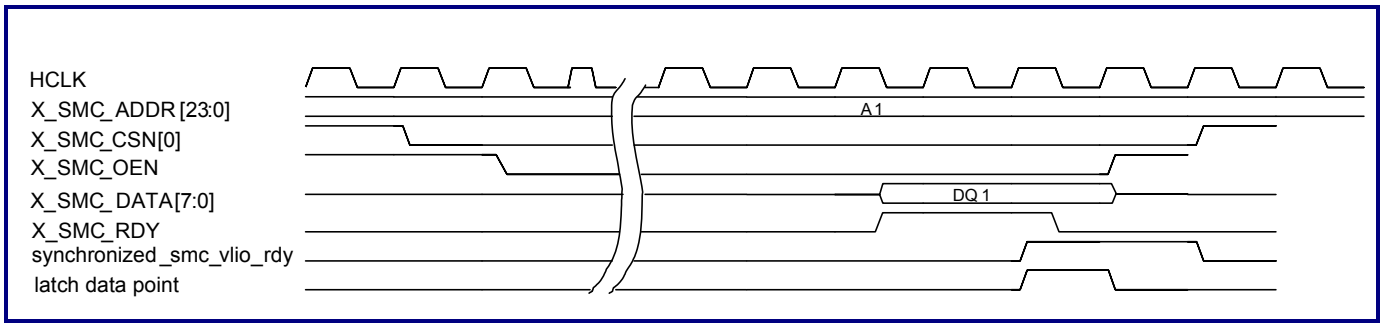



**Figure 8-6. Timing Diagram of SMC Read from VLIO Devices**

### 8.4.5 Shadow Status Register (Offset = 0x40)

Because of the access speed of the boot ROM, most systems need to shadow the fast SRAM to boot the address space of the ROM. The static memory controller proposes a shadow status register for users to switch between Bank0 and other banks. Table 8-16 lists the bit assignments of the shadow status register.

**Table 8-16. Shadow Status Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:9] | - | - | Reserved |
| 8 | SSR_STS | R | Shadow status<br>If this bit is set to '1,' the controller is in the bank switch mode. All the memory accesses to Bank0 will be redirected to the SSR_BNKNUM bit and set by the shadow status register, and all the memory accesses to the BANKNUM bit will be redirected to Bank0. |
| [7:6] | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| 5 | SSR_REQ | R/W/AC | Shadow request<br>If this register is set to ‘1,’ the controller will try to switch the bank information between Bank0 and SSR_BNKNUM. Once complete, this bit will be cleared to ‘0.’ |
| 4 | SSR_REQM | R/W | Shadow request mode<br>If this register is set to ‘1,’ the shadow enable function is requested. Otherwise, the shadow disable function is requested. |
| 3 | - | - | Reserved |
| [2:0] | SSR_BNKNUM | R/W | Shadow bank number<br>If SSR_STS is set to ‘1,’ all memory accesses to SSR_BNKNUM will be redirected to Bank0. |

# Chapter 9

# DDR2 Memory Controller

This chapter contains the following sections:


- 9.1 General Description
- 9.2 Features
- 9.3 Block Diagram
- 9.4 Function Description
- 9.5 Programming Model
- 9.6 Memory Address Table (MA Table)

## 9.1 General Description

The DDR2 memory controller is an DDR2 SDRAM controller compliant with the AMBA 2.0. This controller supports rich types of the DDR2 SDRAMs and uses a DDR2 SDRAM burst length of four to accelerate the speeds of the read and write operations. The memory controller can prefetch the sequential read data for the burst read commands from the AHB masters to improve the transfer efficiency.

## 9.2 Features

- Supports a maximum of 2G bytes for each rank
- Uses DDR2 SDRAM burst length of four to speed up read and write operations
- Enters DDR2 SDRAM self-refresh mode by using a hardware handshake signal or software configuration
- Prefetches the sequential read data for AHB burst read commands
- Supports several types of MA tables corresponding to different sizes and types
- The arbiter provides two arbitration strategies for granting channels:
  - Two-level round-robin arbitration
  - Two-level round-robin with merging-write arbitration
- Supports synchronous/fully asynchronous/mixed clocking modes for AHB slave interface
- Supports hardware configurable SPLIT responses for AHB slave interface
- Supports Read-only/Write-only/Normal modes for AHB slave interface

## 9.3 Block Diagram

Figure 9-1 depicts the functional block diagram of the DDR2 memory controller.

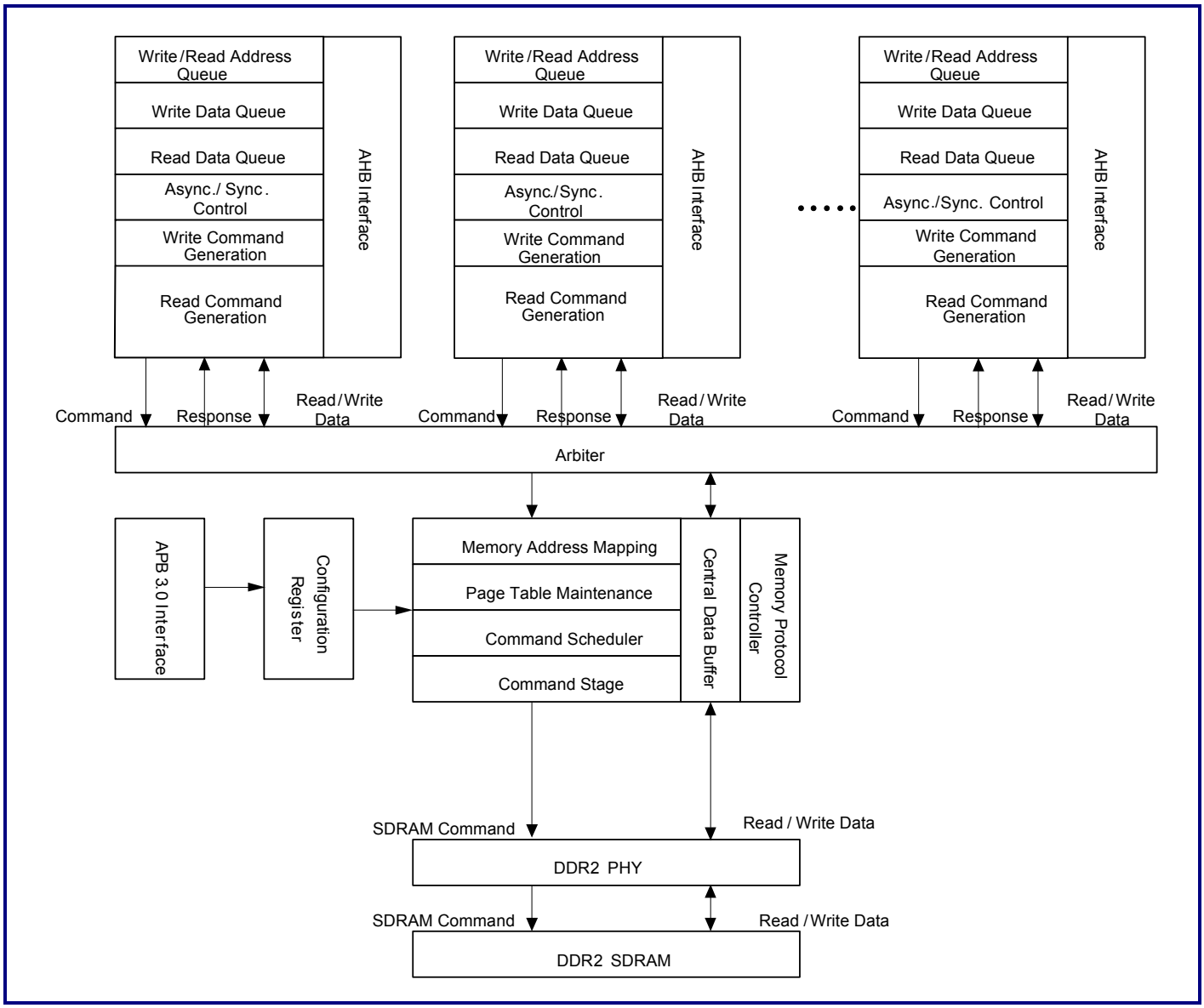



**Figure 9-1. Functional Block Diagram of DDR2 Memory Controller**

## 9.4 Function Description

### 9.4.1 Modes of Operation

The DDR2 controller provides two operation modes: Normal mode and self-refresh mode. When the system is powered on and a hardware is reset, the software needs to initialize the DDR2 controller by programming the control register to enter the normal mode. In the normal mode, all operations run at full speed. In order to save the power, the self-refresh mode is implemented to minimize the power dissipation of the memory module. When bit 2 of the controller register 0X04 (MemCmd_srf) is set to 1'b1, the controller automatically enters the self-refresh mode after all the outstanding commands are finished. The external memory modules will enter the self-refresh mode. To wake up the memory module, users should choose one of the AHB masters to issue the read/write commands to the memory controller. As a consequence, the memory controller will automatically wake up and exit from the self-refresh mode.

### 9.4.2 Refresh Function

The auto-refresh command prevents the data loss in the DDR2 SDRAM. The period of a refresh command can be programmed depending on the speed of the system clock and the DDR2 SDRAM Specification. This value is usually documented in the DDR2 Specification as a specified time interval ($T_{ref}$) in which all rows (*row_num*) should be refreshed completely to prevent the data loss. The average refresh intervals are 7.8 μs (Commercial) and 3.9 μs (Industrial).

### 9.4.3 Channel Arbitration

For the multi-channel configurations, the channel arbiter is responsible for selecting a granted channel which is allowed to issue commands to the memory controller. The arbitration policies are described as below.

#### 9.4.3.1 Two-level Round-robin Arbitration

The two-level round-robin algorithm is used to perform the arbitrations. Based on this algorithm, all channels are divided into two levels, the high-priority level and the low-priority level. Users can specify the high priority channels by setting the register 0x30. For example, five channels are enabled, and channel 2 and channel 4 are specified as the high-priority channels; other channels (0, 1, and 3) are specified as the low-priority channels. The grant sequences will be:

CH2 → CH4 → CH0 → CH2 → CH4 → CH1 → CH2 → CH4 → CH3 → CH2 → CH4 → CH0 ……

Each channel has one 4-bit grant window count (Address offset 0x34). When the granted channel gains the grant and issues one command to the memory controller, its grant window count will be decreased by one. Once the grant window count of the granted channel is decreased to zero, the arbiter will perform the re-arbitration. However, if the granted channel terminates its operation too early, the re-arbitration will be performed even if its grant window count is not zero.

#### 9.4.3.2 Two-level Round-robin with Read-Write Grouping Arbitration

Base on the two-level round-robin algorithm, an advanced mechanism "read–write group" is used to reduce the SDRAM "Write-to-Read" (WtR) wait cycles. The main idea is to merge the same command type (Read with read, write with write) and thus reduce the times of condition "Read back to Write" occurred in the DDR2/DDR3 SDRAM. This reduces the "WtR" wait cycles.

This feature can be enabled by programming the Channel Arbitration Setup register (Address offset 0x30). If the read-write grouping mechanism is enabled, all channels can be classified into four arbitration groups:

1. **HR group:** Channels belonging to the high-priority group are attempting to issue a Read command.
2. **HW group:** Channels belonging to the high-priority group are attempting to issue a Write command.
3. **LR group:** Channels belonging to the low-priority group are attempting to issue a Read command.
4. **LW group:** Channels belonging to the low-priority group are attempting to issue a Write command.

### Channel Re-arbitration

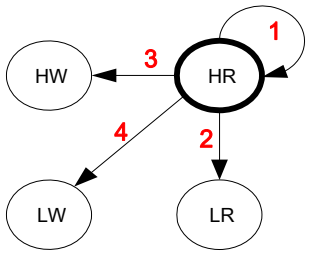
Similar to the two-level round robin arbitration, when the grant window count of a channel becomes zero, the channel re-arbitration will be performed. The arbiter will try to select a channel which belongs to the same arbitration group with current granted channel. For example, channel 0 and channel 2 belong to the LR group; channel 1 belongs to the HW group. If channel 0 is the current granted channel, and channel 1 and channel 2 will keep requesting the arbiter; moreover, the grant window count of channel 0 becomes zero, the arbiter will select channel 2 to be the next granted channel for channel 2 and channel 0 belong to the same group (Low-level channels which issue the read commands).

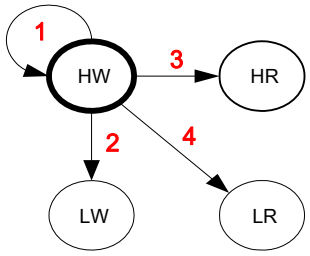
### Group Re-arbitration

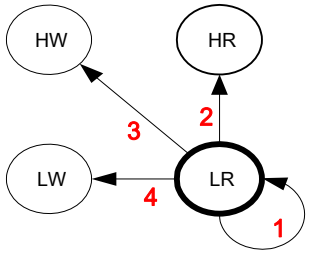
A programmable counter, "group_grant_count" (Offset 0x30), defines the maximum command counts allowed to be issued for a single group. Each time a channel issues a command to the memory controller, the "group_grant_count" will be decreased by one. When the "group_grant_count" becomes zero, the group-level re-arbitration will be performed. The arbiter will try to select an arbitration group as the new active group, and restart the counting of "group_grant_count".

By the way, in order to avoid the arbiter from being stalled in the "HR" and "LR" groups or "HW" and "LW" groups, a counter named "type_grant_count" has the value equals to twice the "group_grant_count". Each time a channel issues a read/write command to the memory controller, "type_grant_count" will be decreased by one. When "type_grant_count" becomes zero, the type-level re-arbitration will be performed. The arbiter will try to select write/read as the new active group, and restart the counting of "type_grant_count". The selection algorithm is depicted in Figure 9-2.

```
case ( current _ group )

  HR  : if(HR _ req  && group _grantcnt != 0  && read_grantcnt !=0)          //1
            stay in HR group  ;
      else if  (LR _ req  && read_grantcnt !=0)        //2
            grant to LR group  ;
      else if  (HW _req)                               //3
            grant to HW group  ;
      else if  (LW _req)                               //4
            grant to LW group  ;
      else
            stay in HR group  ;

  HW  : if(HW _ req  && group _grantcnt != 0  &&  write _grantcnt !=0)        //1
            stay in HW group  ;
       else if  (LW _ req  &&  write _grantcnt !=0)     //2
            grant to LW group  ;
       else if  (HR _req)                               //3
            grant to HR group  ;
       else if  (LR _req)                               //4
            grant to LR group  ;
       else
            stay in HW group  ;

  LR  : if(LR _ req  && group _grantcnt != 0  && read_grantcnt !=0)          //1
            stay in LR group  ;
      else if  (HR _ req  && read_grantcnt !=0)        //2
            grant to HR group  ;
      else if  (HW _req)                               //3
            grant to HW group  ;
      else if  (LW _req)                               //4
            grant to LW group  ;
      else
            stay in LR group  ;

  LW  : if(LW _ req  && group _grantcnt != 0  &&  write _grantcnt !=0)        //1
            stay in LW group  ;
      else if  (HW _ req  &&  write _grantcnt !=0)     //2
            grant to HW group  ;
      else if  (HR _req)                               //3
            grant to HR group  ;
      else if  (LR _req)                               //4
            grant to LR group  ;
      else
            stay in LW group  ;

endcase
```



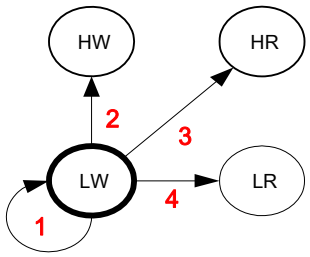




**Figure 9-2. Group-level Re-arbitration**

**Avoidance of Starvation**

Based on the arbitration sequences described in Figure 9-2, the starvation of the low-level groups, LR and LW, may occur if the HR and HW groups continue to request the arbiter. To avoid the starvation conditions, an internal counter, which defines the maximum command counts, is allowed to be issued for the high-level channels. In other words, when a channel that belonged to HR or HW issues a command, this internal counter will be decreased by one. When this counter becomes zero, all the requests from the HR and the HW groups will be masked. The arbiter will be forced to change the grant to the low-level groups (LR and LW).

The reset value of this internal counter is Group Grant Window Count x 4.

#### 9.4.3.3 Cross-channel Data Coherence

In the multi-channel environments, the data coherence may issue between different channels. For example, the channel 0 writes data to address "A", and the channel 1 wants to read the data which was written by channel 0 in address A. Because the DDR2 controller provides the command FIFO and the opportunity for channel switching, it is possible that the read command of channel 1 will exceed the write command of channel 0. Thus, the channel 1 will read the wrong data.

To react to the data hazard situations, the AHB slave interface provides the "Write-flushing mechanism". When the write-flushing is enabled, the DDR2 controller will trigger an interrupt when all commands in command FIFO are popped out to the memory controller.

For the scenario described above (Channel 0 writes and channel 1 reads), the write-flushing commend of channel 0 can be enabled after the write commands are received by the DDR2 controller. When the DDR2 controller triggers an interrupt, it indicates that the write command has already been popped to the memory controller. In other words, channel 1 can start to perform the read access which will never exceed the write command of channel 0. The data coherence between channel 0 and channel 1 will be guaranteed.

## 9.5 Programming Model

Figure 9-3 illustrates the basic timing parameters, which can be configured by setting the register.

R: Read

W: Write

P: Pre-charge

A: Active

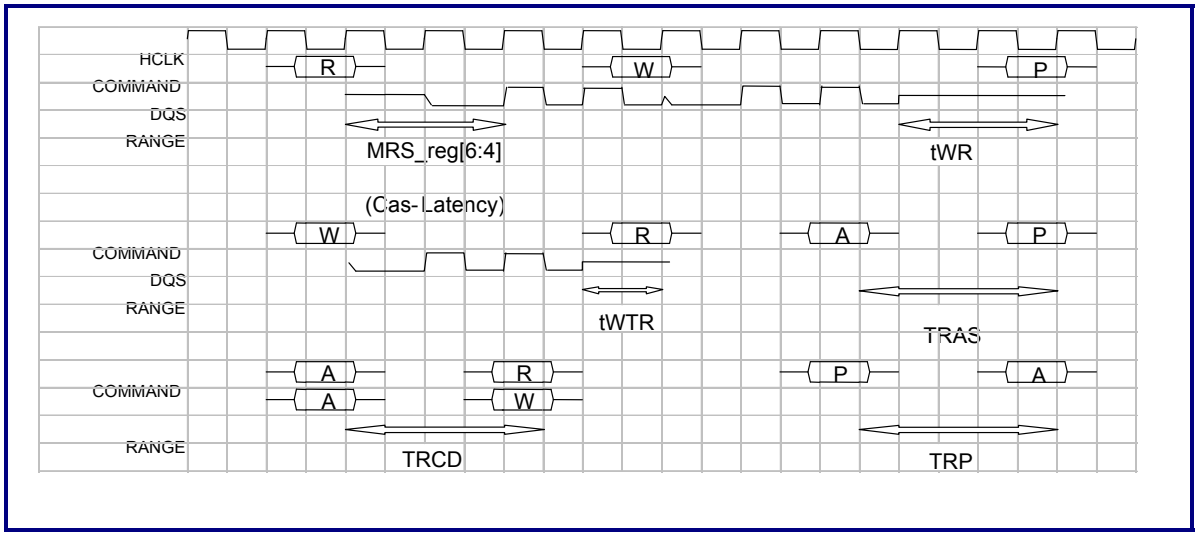



**Figure 9-3. Basic DDR2 RC Timing Parameter**

### 9.5.1 Register Descriptions

The DDR2 controller registers are shown in Table 9-1. The following sections describe the DDR2 controller registers in details.

**Table 9-1. Summary of DDR2 Controller Registers**

| Offset | Type | Description | Reset Value |
|---|---|---|---|
| 0x00 | R/W | Memory controller configuration | 0x00030000 (Pure DDR2)<br>0x00000000 |
| 0x04 | R/W | Memory controller state control | 0x0000c000 |
| 0x08 | R/W | Mode register set value for Mode Register (MR) and Extended Mode Register (EMR) | 0x00000000 |
| 0x0C | R/W | Mode register set value for EMR2 and EMR3 | 0x00000000 |
| 0x10 | R/W | External rank0/rank1 register | 0x00000000 |
| 0x14 | R/W | Timing parameter 0 | 0x00000000 |
| 0x18 | R/W | Timing parameter 1 | 0x00000000 |
| 0x1C | R/W | Timing parameter 2 | 0x00000000 |
| 0x20 | R/W | Data block control | 0x00000041 |
| 0x24 | R/W | DDR PHY read path DLL delay tuning | 8-bit memory width = 0x3<br>16-bit memory width = 0x33<br>32-bit memory width = 0x3333<br>64-bit memory width = 0x33333333 |
| 0x28 | R/W | COMPBLK control | N/A |
| 0x2C | R/W | Automatic power down | 0x00000000 |
| 0x30 | R/W | Channel arbitration setup | 0x10000000 |
| 0x34 | R/W | Channel arbiter grant count - A | 0x07070707 |
| 0x38 | R/W | Channel arbiter grant count - B | 0x07070707 |
| 0x3C | - | Reserved | - |
| 0x40 | R/W | Command flush control | 0x00000000 |
| 0x44 | R/W | Command flush status | 0x00000000 |
| 0x48 | R/W | AHB Split control | 0x00000000 |
| 0xA0 | R/W | AHB INCR read prefetch length 1 | 0x03030303 |
| 0xA4 | R/W | AHB INCR read prefetch length 2 | 0x03030303 |
| 0xA8 | R/W | Initialization waiting cycle count 1 | 0x0001a0ab |

### 9.5.2 Memory Controller Configuration Register (Offset = 0x00)

This configuration register is the DDR2 controller function register. GDS can adjust the read data timing of the DDR2 controller. Please refer to Table 9-2 for more details.

**Table 9-2. Memory Controller Configuration Register (Offset 0x00)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:19] | - | - | Reserved |
| [18] | DisAPBErrResp | R/W | This bit disables the APB error response register.<br>The APB register interface will respond to PSIVERR under one of the following conditions:<br>(1) Write/Read to non-existed registers<br>(2) Write to Read-Only registers<br>Users can set this register to be '1' to disable the APB error response. The default value is '0'. |
| [17:16] | {byone, DDR2_mode} | R/W | 2'b11 = DDR2 SDRAM<br>2'b00 = Reserved<br>2'b01 = Reserved<br>2'b10 = Reserved |
| [15:13] | Post-refresh command counts threshold | R/W | Post-refresh command count threshold<br>The DDR controller will post a refresh command until the counts of the posted refresh commands exceed the threshold.<br>3'b000 = No posted refresh command<br>3'b001 = One posted refresh command threshold<br>3'b010 = Two posted refresh command thresholds<br>3'b011 = Three posted refresh command thresholds<br>3'b100 = Four posted refresh command thresholds<br>3'b101 = Five posted refresh command thresholds<br>3'b110 = Six posted refresh command thresholds<br>3'b111 = Seven posted refresh command thresholds |

| Bit | Name | Type | Description |
|---|---|---|---|
| [12:10] | Auto-Refresh commands in initial SDRAM sequence | R/W | 3'b000 = Not used<br>3'b001 = Two auto-refresh commands<br>3'b010 = Three auto-refresh commands<br>3'b011 = Four auto-refresh commands<br>3'b100 = Five auto-refresh commands<br>3'b101 = Six auto-refresh commands<br>3'b110 = Seven auto-refresh commands<br>3'b111 = Eight auto-refresh commands |
| [9:8] | Memory width | R/W | Memory width<br>2'b00 = 8-bit memory<br>2'b01 = 16-bit memory<br>2'b10 = 32-bit memory<br>2'b11 = 64-bit memory |
| [7:6] | Reserved | - | - |
| [5:4] | DDR2/DDR3 memory address mapping table selection<br>(AMTSEL) | R/W | 2'b00 = The mapping method of the AHB address to the DDR2 SDRAM memory address is RA, BA, and CA.<br>2'b01 = The mapping method of the AHB address to the DDR2 SDRAM memory address is BA, RA, and CA.<br>2'b10 = The mapping method of the Memory Bank Swizzle, the AHB address to the DDR2 SDRAM memory address is RA, CA, BA, and CA.<br>Note: This field only works when the hardware configuration of FTDDR3030_BANK_SWIZZLE_YES is enabled.<br>**AMTSEL = 2'b10 does support the 8-bit memory.**<br>For a 64-bit memory interface, the bank address mapping is inserted between the column addresses.<br>ADDR maps to RA, CA, BA, and CA.<br>Bits[9:8] of the system address are configured as the bank addresses for four banks of the DDR2 SDRAM.<br>Bits[10:8] of the system address are configured as the bank addresses for eight banks of the DDR2 SDRAM.<br>For a 32-bit memory interface, the bank address mapping is inserted between the column addresses.<br>ADDR maps to RA, CA, BA, and CA.<br>Bits[8:7] of the system address are configured as the bank addresses for four banks of the DDR2/DDR3 SDRAM.<br>Bits[9:7] of the system address are configured as the bank addresses for eight banks of the DDR2/DDR3 SDRAM. |

| Bit | Name | Type | Description |
|---|---|---|---|
| | | | For a 16-bit memory interface, the bank address mapping is inserted between the column addresses.<br>ADDR maps to RA, CA, BA, and CA.<br>Bits[7:6] of the system address are configured as the bank addresses for four banks of the DDR2 SDRAM.<br>Bits[8:6] of the system address are configured as the bank addresses for eight banks of the DDR2 SDRAM .<br>2'b11 = Reserved |

### 9.5.3 Memory Controller State Control Register (Offset = 0x04)

Please note that these commands are exclusive. Users should issue one command at a time.

**Table 9-3. Memory Controller State Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | - | - | Reserved |
| [30:18] | - | - | Reserved |
| 17 | - | - | Reserved |
| 16 | - | - | Reserved |
| 15 | AHB command queue empty | RO | 1'b1 = All AHB command queue is empty.<br>1'b0 = All AHB command queue is not empty |
| 14 | Memory controller command queue empty | RO | 1'b1 = Memory controller command queue is empty.<br>1'b0 = Memory controller command queue is not empty. |
| 13 | - | - | Reserved |
| 12 | - | - | Reserved |
| 11 | - | - | Reserved |
| 10 | Self-refresh state | RO | 1'b1 = DDR2 SDRAM is in the self-refresh mode.<br>1'b0 = DDR2 SDRAM is in other state. |
| 9 | Initial State | RO | 1'b1 = DDR2 SDRAM is in the initial state.<br>1'b0 = DDR2 SDRAM is in the initial state. |
| 8 | Initial OK | RO | 1'b1 = DDR2 initial is completed.<br>1'b0 = DDR2 initial is not completed. |
| 7 | - | - | Reserved |
| 6 | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [5:4] | Register mode | R/W | 2'b00 = Mode register<br>2'b01 = Extended mode register (1)<br>2'b10 = Extended mode register (2)<br>2'b11 = Extended mode register (3) |
| 3 | Exit self-refresh command | R/W | This bit is programmed to move the controller of entering the targeted state.<br>When the command is completed, this bit will be cleared to zero.<br>1'b1 = Moving the memory controller from the self-refresh state<br>1'b0 = No effect |
| 2 | Self-refresh command | R/W | This bit is programmed to move the controller of entering the targeted state.<br>When the command is completed, this bit will be cleared to zero.<br>1'b1 = Moving the memory controller to the self-refresh state<br>1'b0 = No effect |
| 1 | MRS command | R/W | This bit is programmed to move the controller of entering the targeted state.<br>When the command is completed, this bit will be cleared to zero.<br>Users should program the MR/EMR/EMRS2/EMRS3 in register offset 0x08 and 0x0C and Offset 0x04 bit[5:4] mode register mode<br>Before starting the MRS command, please:<br>1'b1 = Moving the memory controller to mode register set state<br>1'b0 = No effect |
| 0 | Initial command | R/W | This bit is programmed to move the controller of entering the targeted state.<br>When the command has completed, this bit will be cleared to zero.<br>Setting this bit to 1'b1 will start to perform the DDR2 SDRAM initialization and DDR PHY compensation block calibration.<br>After these tasks are completed, the signal bit[8]of the initial_ok will be set to 1'b1.<br>1'b1 = Moving the memory controller to the DDR2/DDR3 SDRAM initial state<br>1'b0 = No effect |

### 9.5.4 Mode Register Set Value Register of MR and EMR (Offset = 0x08)

This register defines the mode register value and the extended mode register value to be used in the initial sequence. During the initialization of the DDR2 SDRAM, The DDR controller will use the values in this register for the MRS command.

Notes:

1. The mode register content write recovery should be set to the same value of Timing Parameter1 register Offset 0x18 bits[23:20] $t_{WR}$.
2. AL should be set to 0 for the DDR2/DDR3 applications to improve the Write to Read or Read to Write performance.
3. The bit 26 of register offset 0x08 DQS_B and bit 3of register offset 0x020 should be set to the same value. For example, the DDR2 SDRAM and DDR PHY should operate in the same DQS mode, either in the differential mode or single-ended mode.

**Table 9-4. Mode Register and Extended Mode Register Value Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:16] | Extended mode register | R/W | Extended mode register |
| [15:14] | - | - | Reserved |
| [13:0] | Mode register | R/W | Mode register |

### 9.5.5 Mode Register Set Value Register of EMR2 and EMR3 (Offset = 0x0C)

This register defines the mode register value and the extended mode register value to be used in the MRS command after the DDR2 SDRAM initialization.

If users want to modify the mode register value, the offset 0x10 register should be programmed first, and then set bits[2:0] of the offset 0x04 register to 3'b010 (MRS command mode).

**Table 9-5. Extended Mode Register 2 and Extended Mode Register 3 Value Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:16] | Extended mode register 3 | R/W | Extended mode register 3 |
| [15:14] | - | - | Reserved |
| [13:0] | Extended mode register 2 | R/W | Extended mode register 2 |

### 9.5.6 External Rank 0/1 Register (Offset = 0x10)

Please note that the Rank1 base address will be the Rank0 base address + Rank0 size and Rank0 will always be enabled.

**Table 9-6. External Rank0/1 Register (Offset = 0x10)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | RNK0_BASE | R/W | 8-bit base address of external Rank0<br>RNK0 (Chip Select 0) addresses range:<br>{RNK0_BASE, 24'h0} ~ {{RNK0_BASE,24'h0} + RNK0_SIZE – 1'b1}}<br>RNK1 (Chip Select 1) addresses range:<br>{{RNK0_BASE, 24'h0} + RNK0_SIZE } ~ {{RNK0_BASE,24'h0} + RNK0_SIZE + RNK1_SIZE– 1'b1}} |
| [23:17] | - | - | Reserved |
| 16 | RNK1_EN | - | Rank1 enable |
| 15 | - | - | Reserved |
| [14:12] | RNK1_TYPE | R/W | Select the DDR2 type based on the MA table for rank1<br>For example:<br>RAXCAXBA is denoted as 13X10X2.<br>"13X10X2" means that the row address is 13 bits, the column address is 10 bits, and the bank address is 2 bits.<br>3'b000 = 13X9X2<br>3'b001 = 13X10X2<br>3'b010 = 14X10X2<br>3'b011 = 12X10X3<br>3'b100 = 13X10X3<br>3'b101 = 14X10X3<br>3'b110 = 15X10X3<br>3'b111 = Reserved |
| 11 | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [10:8] | RNK1_SIZE | R/W | Memory size of rank1<br>000: 16M bytes<br>001: 32M bytes<br>010: 64M bytes<br>011: 128M bytes<br>100: 256M bytes<br>101: 512M bytes<br>110: 1G bytes<br>111: 2G bytes |
| 7 | - | - | Reserved |
| [6:4] | RNK0_TYPE | R/W | Select the DDR2 type based on the MA table for rank 0<br>For example:<br>RAXCAXBA is denoted as 13X10X2.<br>“13X10X2” means that the row address is 13 bits, the column address is 10 bits, and the bank address is 2 bits.<br>3’b000 = 13X9X2<br>3’b001 = 13X10X2<br>3’b010 = 14X10X2<br>3’b011 = 12X10X3<br>3’b100 = 13X10X3<br>3’b101 = 14X10X3<br>3’b110 = 15X10X3<br>3’b111 = Reserved |
| 3 | - | - | Reserved |
| [2:0] | RNK0_SIZE | R/W | Memory size of rank0<br>000: 16M bytes<br>001: 32M bytes<br>010: 64M bytes<br>011: 128M bytes<br>100: 256M bytes<br>101: 512M bytes<br>110: 1G bytes<br>111: 2G bytes |

### 9.5.7 Timing Parameter 0 Register (Offset = 0x14)

Please note that all parameters are set in terms of the "DDR2CLK/2" cycles. The programmable value of 0 is invalid.

**Table 9-7. Timing Parameter 0 Register (Offset = 0x14)**

| Bits | Name | Type | Description |
|---|---|---|---|
| [31:24] | TRFC | R/W | REFRESH-to-ACTIVE/REFRESH command period<br>8'h0 = Not used<br>8'h1 = 2T<br>8'h2 = 3T<br>...<br>8'hff = 256 T |
| [23:21] | - | - | Reserved |
| [20:16] | TFAW | R/W | Four bank ACTIVE times<br>5'h0 = Not used<br>5'h1 = 2T<br>5'h2 = 3T<br>...<br>5'h1f = 32T |
| [15:14] | - | - | Reserved |
| [13:8] | TRC | R/W | ACTIVE-to-ACTIVE command period for the same Rank<br>6'h0 = Not used<br>6'h1 = 2T<br>6'h2 = 3T<br>...<br>6'h3f = 64T |
| [7:5] | - | - | Reserved |
| [4:0] | TRAS | R/W | ACTIVE-to-PRECHARGE period<br>This parameter specifies the minimum period of an opened row.<br>5'h0 = Not used<br>5'h1 = 2T<br>5'h2 = 3T<br>…<br>5'h1f = 32T |

### 9.5.8 Timing Parameter 1 Register (Offset = 0x18)

Please note that all parameters are set in terms of the "DDR2CLK/2" cycles.

**Table 9-8. Timing Parameter 1 Register (Offset = 0x18)**

| Bits | Name | Type | Description |
|---|---|---|---|
| 31 | - | - | Reserved |
| [30:28] | TWTR | R/W | Internal WRITE to READ delay<br>READ command after the last write data delay cycle<br>3'h0 = Not used<br>3'h1 = 2T<br>3'h2 = 3T<br>...<br>3'h7 = 8T |
| 27 | - | - | Reserved |
| [26:24] | TRTP | R/W | Internal READ to PRECHARGE delay<br>3'h0 = Not used<br>3'h1 = 2T<br>3'h2 = 3T<br>...<br>3'h7 = 8T |
| [23:20] | TWR | R/W | Write recovery time<br>WRITE to PRECHARGE<br>The last non-write data cycle at the MCLK rising edge to the PRECHARGE command cycle<br>4'h0 = Not used<br>4'h1 = 2T<br>4'h2 = 3T<br>...<br>4'hf = 16T |

| Bits | Name | Type | Description |
|---|---|---|---|
| [19:16] | TMOD | R/W | Mode register set command update delay<br>4'h0 = Not used<br>4'h1 = 2T<br>4'h2 = 3T<br>...<br>4'hf = 16T |
| [15:12] | TMRD | R/W | Cycle time of load mode register command<br>4'h0 = Not used<br>4'h1 = 2T<br>4'h2 = 3T<br>4'h3 = 4T<br>...<br>4'hf = 16T |
| [11:8] | TRP | R/W | PRECHARGE period<br>4'h0 = Not used<br>4'h1 = 2T<br>4'h2 = 3T<br>...<br>4'hf = 16T |
| [7:4] | TRRD | R/W | ACTIVE to ACTIVE command period for different banks<br>4'h0 = Not used<br>4'h1 = 2T<br>4'h2 = 3T<br>...<br>4'hf = 16T |
| [3:0] | TRCD | R/W | Minimum delay between the ACTIVE and READ/WRITE commands<br>4'h0 = Not used<br>4'h1 = 2T<br>4'h2 = 3T<br>...<br>4'hf = 16T<br>Note: If AL is not set as '0', the READ/WRITE command will be directly followed by the ACTIVE command. If AL is set as '0', the ACTIVE to READ/WRITE command will have the TRCD timing constraints. |

### 9.5.9 Timing Parameter 2 Register (Offset = 0x1C)

**Table 9-9. Timing Parameter 2 Register (Offset = 0x1C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:30] | TWtoR_ctrl | R/W | Controller specific<br>Additional delay cycles for the WRITE command to the READ command<br>The programmed value is to add delay between the write and the read command.<br>2'h0 = 0T<br>2'h1 = 1T<br>2'h2 = 2T<br>2'h3 = 3T |
| [29:28] | TWtoW_ctrl | R/W | Controller specific<br>Additional delay cycles for the WRITE command to the WRITE command<br>For example: Different ranks<br>The programmed value is to compensate the unbalanced trace or I/O between different ranks of DDR2 SDRAM to avoid the DQS driving contention.<br>2'h0 = 0T<br>2'h1 = 1T<br>2'h2 = 2T<br>2'h3 = 3T |
| [27:26] | TRtoR_ctrl | R/W | Controller specific<br>Additional delay cycles for the READ command to the READ command<br>For example: Different ranks<br>The programmed value is to compensate the unbalanced trace or I/O between different ranks of DDR2 SDRAM to avoid the DQS driving contention.<br>2'h0 = 0T<br>2'h1 = 1T<br>2'h2 = 2T<br>2'h3 = 3T |

| Bit | Name | Type | Description |
|---|---|---|---|
| [25:24] | TRtoW_ctrl | R/W | Controller specific<br>Additional delay cycles for the READ command to the WRITE command<br>The DQS will be driven by different devices in the read and write operations. The programmed value will compensate the round-trip delay between CHIP and DDR2 SDRAM to avoid the DQS driving contention.<br>Note: This value should be set as 2'h2 or 2'h3.<br>2'h0 = 0T<br>2'h1 = 1T<br>2'h2 = 2T<br>2'h3 = 3T |
| [23:16] | - | - | Reserved |
| [15:8] | TXSR | R/W | Exit from the self-refresh mode to a command period<br>This should be programmed as the maximum of $t_{XSNR}$ and $t_{XSRD}$<br>8'h0 = Not used<br>8'h1 = 1 X 8T<br>8'h2 = 2 X 8T<br>...<br>8'hff = 255 X 8T |
| [7:0] | TREFI | R/W | Average periodic refresh interval<br>One refresh command should be issued if the refresh counter equals to the refresh interval. Please note that bit 7 is only for the selection between x32T and x8T.<br>**bit 7 = 1'b0:**<br>8'b0_000_0000 = Not used<br>8'b0_000_0001 = 1X32T<br>8'b0_000_0010 = 2X32T<br>...<br>8'b0_111_1111 = 127X32T<br>**bit 7 = 1'b1:**<br>8'b1_000_0000 = 0x8T<br>8'b1_000_0001 = 1X8T<br>8'b1_000_0010 = 2X8T<br>...<br>8'b1_111_1111 = 127X8T |

### 9.5.10 DDR2 PHY Command and Data Block Control Register (Offset = 0x20)

**Table 9-10. DDR2 PHY Command and Data Block Control Register (Offset = 0x20)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | fldo_byte7 | RO | Byte7 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at the lock state. |
| 30 | fldo_byte6 | RO | Byte6 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at the lock state. |
| 29 | fldo_byte5 | RO | Byte5 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at the lock state. |
| 28 | fldo_byte4 | RO | Byte4 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at the lock state. |
| 27 | fldo_byte3 | RO | Byte3 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at the lock state. |
| 26 | fldo_byte2 | RO | Byte2 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at lock state. |
| 25 | fldo_byte1 | RO | Byte1 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at the lock state. |
| 24 | fldo_byte0 | RO | Byte0 DLL lock flag indicates that DLL is at the lock state.<br>1'b1 = DLL is at the lock state.<br>1'b0 = DLL is not at the lock state. |
| [23:17] | - | - | Reserved |
| 16 | fldo_cmd | RO | DDR3 PHY only<br>1'b1 = Command block DLL is at the lock state.<br>1'b0 = Command block DLL is not at the lock state. |
| [15:14] | - | - | Reserved |
| 13 | MDDR | R/W | Note: This port does not exist in the IP version 1.0.0. |
| 12 | DDR1 | R/W | Note: This port does not exist in the IP version 1.0.0. |
| 11 | dqie | R/W | DQ and DQS receiver enable |

| Bit | Name | Type | Description |
|---|---|---|---|
| | | | 1’b1 = Enable<br>1’b0 = Disable<br>Note: In the automatic power-down mode, the dqie bit will be controlled by hardware if auto_ioctrl_pdn is set to 1’b1. |
| 10 | clkoen | R/W | The output control of DDR PHY CK, CKB, and ODT driver<br>1’b1 = Enable<br>1’b0 = Disable |
| 9 | cmdaddroen | R/W | The output control of DDR PHY ADDR, CK, CKB, BA, RAS, CAS, WE and CS driver<br>1’b1 = Enable<br>1’b0 = Disable<br>Note: In the automatic power-down mode, the cmdaddroen bit will be controlled by hardware if auto_ioctrl_pdn is set to 1’b1. |
| 8 | odtoen | R/W | Enable output drivers of ODT[1:0]<br>‘1’b1’: Enable<br>‘1’b0’: Disable<br>Note: In the automatic power-down mode, the odtoen bit will be controlled by hardware if auto_ioctrl_pdn is set to 1’b1. |
| 7 | auto_ioctrl_pdn | R/W | Automatic control of the I/O output buffer at power-down state<br>1’b1 = Enable<br>1’b0 = Disable<br>When the automatic control is enabled, the odtoen, cmdaddroen, and dqie bits will be disabled at the automatic power-down and self-refresh states.<br>The clkoen bit is disabled at the self-refresh state. |
| 6 | IO18V | R/W | DDR3 PHY only<br>1’b1 = I/O is 1.8 V.<br>1’b0 = I/O is 1.5 V. |
| 5 | dqslowfen | R/W | Enable the DQS bypass DLL<br>This bit is used for the low-frequency operation to control the DDR PHY bypass DLL. |
| 4 | dmyodten | R/W | On-die termination enable for the DDR PHY DUMMY pad of the data block |
| 3 | sio | R/W | This bit should be compatible with the DQS_B enable bit in the extended mode register.<br>1’b0 = DQS is in the differential mode.<br>1’b1 = DQS is in the single-ended mode. |

| Bit | Name | Type | Description |
|---|---|---|---|
| [2:0] | odtmd[2:0] | R/W | Set the ODT value of DDR2 PHY, only odtmd[1:0] is used for DDR2 PHY<br>3'b010: ODT = 150 Ω<br>3'b001: ODT = 75 Ω<br>3'b011: ODT = 50 Ω<br>3'b000: ODT disabled<br><br>Set the ODT value of DDR2/DDR3 Combo PHY<br>3'b111: ODT = 20 Ω<br>3'b110: ODT = 24 Ω<br>3'b101: ODT = 30 Ω<br>3'b100: ODT = 40 Ω<br>3'b011: ODT = 40 Ω<br>3'b010: ODT = 60 Ω<br>3'b001: ODT = 120 Ω<br>3'b000: ODT disabled. |

### 9.5.11 DDR PHY Read Path DLL Delay Tuning Register (Offset = 0x24)

dllsel is used for adjusting the read DQ and read DQS. The read DQS should be in the center of the DQ data eye for more timing margin. Please refer to the corresponding DDR PHY Application Note for more detailed information.

Note: dllsel_byte0, dllsel_byte2, dllsel_byte4, and dllsel_byte6 are used in the 16-bit DDR2 PHY.

**Table 9-11. DDR PHY Read Path DLL Delay Tuning Register (Offset = 0x24)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | - | - | Reserved |
| [30:28] | dllsel_byte7 | R/W | Read DQS fine tuning of byte7 to the center of the DQ data eye |
| 27 | - | - | Reserved |
| [26:24] | dllsel_byte6 | R/W | Read DQS fine tuning of byte6 to the center of the DQ data eye |
| 23 | - | - | Reserved |
| [22:20] | dllsel_byte5 | R/W | Read DQS fine tuning of byte5 to the center of the DQ data eye |
| 19 | - | - | Reserved |
| [18:16] | dllsel_byte4 | R/W | Read DQS fine tuning of byte4 to the center of the DQ data eye |
| 15 | - | - | Reserved |
| [14:12] | dllsel_byte3 | R/W | Read DQS fine tuning of byte3 to the center of the DQ data eye |
| 11 | - | - | Reserved |
| [10:8] | dllsel_byte2 | R/W | Read DQS fine tuning of byte2 to the center of the DQ data eye |
| 7 | - | - | Reserved |
| [6:4] | dllsel_byte1 | R/W | Read DQS fine tuning of byte1 to the center of the DQ data eye |
| 3 | - | - | Reserved |
| [2:0] | dllsel_byte0 | R/W | Read DQS fine tuning of byte0 to the center of the DQ data eye |

### 9.5.12 COMPBLK Control Register (Offset = 0x28)

Please refer to the corresponding DDR PHY Application Note for more detailed information.

**Table 9-12. COMPBLK Control Register (Offset = 0x28)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:25] | - | - | Reserved |
| [24:19] | DON | RO | COMPBLK pull-down NMOS impedance value |
| [18:13] | DOP | RO | COMPBLK pull-up PMOS impedance value |
| [12:7] | DIN | R/W | Pre-set value for COMPBLK pull-down NMOS impedance |
| [6:1] | DIP | R/W | Pre-set value for COMPBLK pull-up PMOS impedance |
| 0 | COMP_SEL | R/W | Compensation method selection pin<br>1'b1 = Enable COMPBLK calibration before the DDR2/DDR3 initial<br>1'b0 = Use the preset values of DIP and DIN |

### 9.5.13 Automatic Power-down Control Register (Offset = 0x2C)

**Table 9-13. Automatic Power-down Control Register (Offset = 0x2C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:13] | - | - | Reserved |
| 12 | Enable | R/W | Automatic power-down enable<br>1'b1 = Enable automatic power-down<br>1'b0 = Disable automatic power-down |
| [11:0] | Timeout | R/W | Automatic power-down timeout<br>When the command queue is empty, the automatic power-down counter will start counting down. When the counter is zero, CKE will be automatically pulled low.<br><br>12'h000 = Not used<br>12'h001 = 1 x 4T = 4T<br>12'h002 = 2 x 4T = 8T<br>…<br>12'hfff = 4095 x 4T = 16380T |

### 9.5.14 Channel Arbitration Setup Register (Offset = 0x30)

This register provides the parameter setup for the channel arbitration.

**Table 9-14. Channel Arbitration Setup Register (Offset = 0x30)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | RW_Grp_EN | - | Enable read-write grouping arbitration<br>Set this bit to '1' to enable the read-write-grouping arbitration. Otherwise, the normal two-level round-robin arbitration will be used. |
| [30:29] | - | - | Reserved |
| [28:24] | Grant_Cnt_Group | R/W | For the read-write grouping arbitration, this field sets the initial value of the group grant window count.<br>This field is not used if the read-write-grouping arbitration is disabled. |
| [23:8] | - | - | Reserved |
| [7:0] | Ch_hi_prior | R/W | All channels are divided into two priority levels:High-priority level andlow-priority level. These bits indicate which channel belongs to the high-priority level. Setting each bit to 1'b1 means that the corresponding channel belongs to the high-priority level.<br>Bit[0] for channel 0<br>Bit[1] for channel 1<br>Bit[2] for channel 2<br>Bit[3] for channel 3<br>Bit[4] for channel 4<br>Bit[5] for channel 5<br>Bit[6] for channel 6<br>Bit[7] for channel 7 |

### 9.5.15 Channel Arbiter Grant Count Register – A (Offset = 0x34)

Each channel has its own grant window counter. The value set in this register will be loaded into the grant window counters of channel 0, channel 1, channel 2, and channel 3, when the granted channel is changed by arbiter. Once the granted channel pops a command to the memory controller, the grant window counter of this channel will be decreased by one. If the grant window counter of the current grated channel becomes 0, the arbiter will switch the grant to other channels. The grant window counter of each channel defines the maximum service command counts allowed for each channel when that channel is granted by the arbiter.

Please note that a command indicates a DDR2 WRAP4 R/W command.

**Table 9-15. Channel Arbiter Grant Count Register – A (Offset = 0x34)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | - | R/W | Reserved |
| [28:24] | ARB_CNT3 | R/W | The maximum allowed commands to be issued once channel 3 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |
| [23:21] | - | R/W | Reserved |
| [20:16] | ARB_CNT2 | R/W | The maximum allowed commands to be issued once channel 2 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |
| [15:13] | - | R/W | Reserved |
| [12:8] | ARB_CNT1 | R/W | The maximum allowed commands to be issued once channel 1 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |
| [7:5] | - | R/W | Reserved |
| [4:0] | ARB_CNT0 | R/W | The maximum allowed commands to be issued once channel 0 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |

### 9.5.16 Channel Arbiter Grant Count Register – B (Offset = 0x38)

Similar to register offset 0x34, this register defines the maximum service command counts of channel 4, channel 5, channel 6, and channel 7.

**Table 9-16. Channel Arbiter Grant Count Register – B (Offset = 0x38)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | - | R/W | Reserved |
| [28:24] | ARB_CNT7 | R/W | The maximum allowed commands to be issued once channel 7 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |
| [23:21] | - | R/W | Reserved |
| [20:16] | ARB_CNT6 | R/W | The maximum allowed commands to be issued once channel 6 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |
| [15:13] | - | R/W | Reserved |
| [12:8] | ARB_CNT5 | R/W | The maximum allowed commands to be issued once channel 5 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |
| [7:5] | - | R/W | Reserved |
| [4:0] | ARB_CNT4 | R/W | The maximum allowed commands to be issued once channel 4 is granted.<br>0 = 2 commands<br>1 = 3 commands<br>2 = 4 commands<br>…<br>31 = 33 commands |

### 9.5.17 Command Flush Control Register (Offset = 0x40)

**Table 9-17. Command Flush Control Register (Offset = 0x40)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | flush_int_en | R/W | Interrupt enable for the command flushing<br>When the command flushing is completed, FTDDR3030 will trigger an interrupt if the corresponding bit of this field is set to '1'.<br>Bit[8] for channel 0<br>Bit[9] for channel 1<br>Bit[10] for channel 2<br>Bit[11] for channel 3<br>Bit[12] for channel 4<br>Bit[13] for channel 5<br>Bit[14] for channel 6<br>Bit[15] for channel 7 |
| [7:0] | cmd_flush_en | R/W | Enable of command flushing<br>Set the corresponding bit of this field to '1' to start the command flushing<br>Bit[0] for channel 0<br>Bit[1] for channel 1<br>Bit[2] for channel 2<br>Bit[3] for channel 3<br>Bit[4] for channel 4<br>Bit[5] for channel 5<br>Bit[6] for channel 6<br>Bit[7] for channel 7<br>When the command flushing is completed, the corresponding bit of this field will be automatically cleared to '0' by FTDDR3030. |

### 9.5.18 Command Flush Status Register (Offset = 0x44)

**Table 9-18. Command Flush Status Register (Offset = 0x44)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | flush_done | R/W1C | Status of command flushing<br>This field records the status of the command flushing for each channel. When a channel finished the command flushing, the corresponding bit of this field will be set to '1' by FTDDR3030.<br>Bit[0] for channel 0<br>Bit[1] for channel 1<br>Bit[2] for channel 2<br>Bit[3] for channel 3<br>Bit[4] for channel 4<br>Bit[5] for channel 5<br>Bit[6] for channel 6<br>Bit[7] for channel 7<br>This field should be cleared by writing '1' to the corresponding bit. |

### 9.5.19 AHB SPLIT Control Register (Offset = 0x48)

If the AHB channels are configured with supporting SPLIT response, the SPLIT function can be disabled by setting the corresponding bits in this register.

**Table 9-19. AHB SPLIT Control Register (Offset = 0x48)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Not used for this version, must be 0. |
| 7 | CH7_split_disable | R/W | This bit works only when channel 7 is enabled and is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 7 |
| 6 | CH6_split_disable | R/W | This bit works only when channel 6 is enabled and is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 6 |
| 5 | CH5_split_disable | R/W | This bit works only when channel 5 is enabled and is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 5 |

| Bit | Name | Type | Description |
|---|---|---|---|
| 4 | CH4_split_disable | R/W | This bit works only when channel 4 is enabled and is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 4 |
| 3 | CH3_split_disable | R/W | This bit works only when channel 3 is enabled and is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 3 |
| 2 | CH2_split_disable | R/W | This bit works only when channel 2 is enabled and is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 2 |
| 1 | CH1_split_disable | R/W | This bit works only when channel 1 is enabled and is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 1 |
| 0 | CH0_split_disable | R/W | This bit works only when channel 0 is configured as an AHB channel with the SPLIT response.<br>Set this bit to '1' to disable the SPLIT response of channel 0 |

### 9.5.20 AHB INCR Read Pre-fetch Length 1 (Offset = 0xA0)

FTDDR3030 provides two mechanisms to prefetch read data for AHB unspecified-length (INCR) read bursts: Limited prefetching and unlimited prefetching.

For the limited prefetching, users define a value that is used as a "total number" of read commands. FTDDR3030 generates the DDR read commands for this number when it receives an AHB INCR read burst. For example, if the user-defined value is *N*, FTDDR3030 will generate three *N*+1 DDR read commands to prefetch the read data for any AHB INCR bursts.

For the unlimited prefetching, users define a value that is used as a "speed factor" of read commands. In this mode, FTDDR3030 will keep generating the read commands until it receives an AHB early termination condition. The user-defined value determines the maximum number of prefetch read commands that are on-going in pipeline. For example, if the user-defined value is *N*, FTDDR3030 will first generate the *N*+1 DDR read commands and will then suspend the command generation until the on-going command count reaches *N*+1. When the data belonged to the first read command arrives to read FIFO, FTDDR3030 will revive to generate the fourth read command if the on-going command count is less than *N*+1. This mechanism will be repeatedly performed until the early termination of AHB occurs.

**Table 9-20. AHB INCR Read Pre-fetch Length 1 (Offset = 0xA0)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | CH3_limited_pref | R/W | Prefetch mechanism of channel 3<br>1: Limited prefetching<br>0: Unlimited prefetching |
| [30:29] | - | - | Reserved |
| [28:24] | CH3_pref_value | R/W | User-defined value of channel 3 for INCR read prefetching<br>0: One command<br>1: Two commands<br>2: Three commands<br>And so on |
| 23 | CH2_limited_pref | R/W | Prefetch mechanism of channel 2<br>1: Limited prefetching<br>0: Unlimited prefetching |
| [22:21] | - | - | Reserved |
| [20:16] | CH2_pref_value | R/W | User-defined value of channel 2 for INCR read prefetching<br>0: One command<br>1: Two commands<br>2: Three commands<br>And so on |
| 15 | CH1_limited_pref | R/W | Prefetch mechanism of channel 1<br>1: Limited prefetching<br>0: Unlimited prefetching |
| [14:13] | - | - | Reserved |
| [12:8] | CH1_pref_value | R/W | User-defined value of channel 1 for INCR read prefetching<br>0: One command<br>1: Two commands<br>2: Three commands<br>And so on |
| 7 | CH0_limited_pref | R/W | Prefetch mechanism of channel 0<br>1: Limited prefetching<br>0: Unlimited prefetching |
| [6:5] | - | - | Reserved |
| [4:0] | CH0_pref_value | R/W | User-defined value of channel 0 for the INCR read prefetching<br>0: One command |

| Bit | Name | Type | Description |
|---|---|---|---|
| | | | 1: Two commands |
| | | | 2: Three commands |
| | | | And so on |

## 9.5.21 AHB INCR Read Pre-fetch Length 2 (Offset = 0xA4)

**Table 9-21. AHB INCR Read Pre-fetch Length 2 (Offset = 0xA4)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | CH7_limited_pref | R/W | Prefetch mechanism of channel 7 |
| | | | 1: Limited prefetching |
| | | | 0: Unlimited prefetching |
| [30:29] | - | - | Reserved |
| [28:24] | CH7_pref_value | R/W | User-defined value of channel 7 for INCR read prefetching |
| | | | 0: One command |
| | | | 1: Two commands |
| | | | 2: Three commands |
| | | | And so on |
| 23 | CH6_limited_pref | R/W | Prefetch mechanism of channel 6 |
| | | | 1: Limited prefetching |
| | | | 0: Unlimited prefetching |
| [22:21] | - | - | Reserved |
| [20:16] | CH6_pref_value | R/W | User-defined value of channel 6 for INCR read prefetching |
| | | | 0: One command |
| | | | 1: Two commands |
| | | | 2: Three commands |
| | | | And so on |
| 15 | CH5_limited_pref | R/W | Prefetch mechanism of channel 5 |
| | | | 1: Limited prefetching |
| | | | 0: Unlimited prefetching |
| [14:13] | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [12:8] | CH5_pref_value | R/W | User-defined value of channel 5 for INCR read prefetching<br>0: One command<br>1: Two commands<br>2: Three commands<br>And so on |
| 7 | CH4_limited_pref | R/W | Prefetch mechanism of channel 4<br>1: Limited prefetching<br>0: Unlimited prefetching |
| [6:5] | | | Reserved |
| [4:0] | CH4_pref_value | R/W | User-defined value of channel 4 for INCR read prefetching<br>0: One command<br>1: Two commands<br>2: Three commands<br>And so on |

### 9.5.22 Initialization of Waiting Cycle Count 1 (Offset = 0xA8)

**Table 9-22. Initialization of Waiting Cycle Count 1 (Offset = 0xA8)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:20] | - | - | Reserved |
| [19:0] | wait_cycle_200us | R/W | After powering up the ramp of SDRAM, a waiting time of 200 µs is required before the RESET signal becomes inactive. This register defines the waiting clock cycles of FTDDR3030 to satisfy the requirement of 200 µs. |

## 9.6 Memory Address Table (MA Table)

The DDR2 controller supports various types of DDR2 SDRAMs. Users must refer to the MA tables (From Table 9-23 through Table 9-33) for the external rank0 configuration register, RNK_TYPE.

Each RANK must choose a correct type from the MA tables. The DDR2 controller supports two types of the MA tables, which can be configured through the AMTSEL bit of the configuration register (Offset 0x00).

**Table 9-23. MA Table (AMTSEL = 2'b00) in 8-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MA1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MA2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA9 | - | 9 | 9 | 9 | 9 | 9 | 9 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 9 | 10 | 10 | 10 | 10 | 10 | 10 |
| BA1 | 10 | 11 | 11 | 11 | 11 | 11 | 11 |
| BA2 | - | - | - | 12 | 12 | 12 | 12 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 11 | 12 | 12 | 13 | 13 | 13 | 13 |
| MA1 | 12 | 13 | 13 | 14 | 14 | 14 | 14 |
| MA2 | 13 | 14 | 14 | 15 | 15 | 15 | 15 |
| MA3 | 14 | 15 | 15 | 16 | 16 | 16 | 16 |
| MA4 | 15 | 16 | 16 | 17 | 17 | 17 | 17 |
| MA5 | 16 | 17 | 17 | 18 | 18 | 18 | 18 |
| MA6 | 17 | 18 | 18 | 19 | 19 | 19 | 19 |
| MA7 | 18 | 19 | 19 | 20 | 20 | 20 | 20 |
| MA8 | 19 | 20 | 20 | 21 | 21 | 21 | 21 |
| MA9 | 20 | 21 | 21 | 22 | 22 | 22 | 22 |
| MA10 | 21 | 22 | 22 | 23 | 23 | 23 | 23 |
| MA11 | 22 | 23 | 23 | 24 | 24 | 24 | 24 |
| MA12 | 23 | 24 | 24 | - | 25 | 25 | 25 |
| MA13 | - | - | 25 | - | - | 26 | 26 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA14 | - | - | - | - | - | - | 27 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-24. MA Table (AMTSEL = 2'b00) in 16-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MA1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA8 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| MA9 | - | 10 | 10 | 10 | 10 | 10 | 10 |
| MA10 | - | - | - | - | - | - | |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 10 | 11 | 11 | 11 | 11 | 11 | 11 |
| BA1 | 11 | 12 | 12 | 12 | 12 | 12 | 12 |
| BA2 | - | - | - | 13 | 13 | 13 | 13 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 12 | 13 | 13 | 14 | 14 | 14 | 14 |
| MA1 | 13 | 14 | 14 | 15 | 15 | 15 | 15 |
| MA2 | 14 | 15 | 15 | 16 | 16 | 16 | 16 |
| MA3 | 15 | 16 | 16 | 17 | 17 | 17 | 17 |
| MA4 | 16 | 17 | 17 | 18 | 18 | 18 | 18 |
| MA5 | 17 | 18 | 18 | 19 | 19 | 19 | 19 |
| MA6 | 18 | 19 | 19 | 20 | 20 | 20 | 20 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA7 | 19 | 20 | 20 | 21 | 21 | 21 | 21 |
| MA8 | 20 | 21 | 21 | 22 | 22 | 22 | 22 |
| MA9 | 21 | 22 | 22 | 23 | 23 | 23 | 23 |
| MA10 | 22 | 23 | 23 | 24 | 24 | 24 | 24 |
| MA11 | 23 | 24 | 24 | 25 | 25 | 25 | 25 |
| MA12 | 24 | 25 | 25 | - | 26 | 26 | 26 |
| MA13 | - | - | 26 | - | - | 27 | 27 |
| MA14 | - | - | - | - | - | - | 28 |
| MA15 | - | | - | - | - | - | - |

**Table 9-25. MA Table (AMTSEL = 2'b00) in 32-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA5 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA6 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA7 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| MA8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| MA9 | - | 11 | 11 | 11 | 11 | 11 | 11 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 11 | 12 | 12 | 12 | 12 | 12 | 12 |
| BA1 | 12 | 13 | 13 | 13 | 13 | 13 | 13 |
| BA2 | | - | - | 14 | 14 | 14 | 14 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| TYPE | 13X9X2 | 13X10X2 | 14X10X2 | 12X10X3 | 13X10X3 | 14X10X3 | 15X10X3 |
| SELECT | 3'b000 | 3'b001 | 3'b010 | 3'b011 | 3'b100 | 3'b101 | 3'b110 |
| Row | RA | RA | RA | RA | RA | RA | RA |
| MA0 | 13 | 14 | 14 | 15 | 15 | 15 | 15 |
| MA1 | 14 | 15 | 15 | 16 | 16 | 16 | 16 |
| MA2 | 15 | 16 | 16 | 17 | 17 | 17 | 17 |
| MA3 | 16 | 17 | 17 | 18 | 18 | 18 | 18 |
| MA4 | 17 | 18 | 18 | 19 | 19 | 19 | 19 |
| MA5 | 18 | 19 | 19 | 20 | 20 | 20 | 20 |
| MA6 | 19 | 20 | 20 | 21 | 21 | 21 | 21 |
| MA7 | 20 | 21 | 21 | 22 | 22 | 22 | 22 |
| MA8 | 21 | 22 | 22 | 23 | 23 | 23 | 23 |
| MA9 | 22 | 23 | 23 | 24 | 24 | 24 | 24 |
| MA10 | 23 | 24 | 24 | 25 | 25 | 25 | 25 |
| MA11 | 24 | 25 | 25 | 26 | 26 | 26 | 26 |
| MA12 | 25 | 26 | 26 | - | 27 | 27 | 27 |
| MA13 | - | - | 27 | - | - | 28 | 28 |
| MA14 | - | - | - | - | - | - | 29 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-26. MA Table (AMTSEL = 2'b00) in 64-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| TYPE | 13X9X2 | 13X10X2 | 14X10X2 | 12X10X3 | 13X10X3 | 14X10X3 | 15X10X3 |
| SELECT | 3'b000 | 3'b001 | 3'b010 | 3'b011 | 3'b100 | 3'b101 | 3'b110 |
| Column | CA | CA | CA | CA | CA | CA | CA |
| MA0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA3 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA4 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA5 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA6 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA7 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| MA8 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| MA9 | - | 12 | 12 | 12 | 12 | 12 | 12 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 12 | 13 | 13 | 13 | 13 | 13 | 13 |
| BA1 | 13 | 14 | 14 | 14 | 14 | 14 | 14 |
| BA2 | | | | 15 | 15 | 15 | 15 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 14 | 15 | 15 | 16 | 16 | 16 | 16 |
| MA1 | 15 | 16 | 16 | 17 | 17 | 17 | 17 |
| MA2 | 16 | 17 | 17 | 18 | 18 | 18 | 18 |
| MA3 | 17 | 18 | 18 | 19 | 19 | 19 | 19 |
| MA4 | 18 | 19 | 19 | 20 | 20 | 20 | 20 |
| MA5 | 19 | 20 | 20 | 21 | 21 | 21 | 21 |
| MA6 | 20 | 21 | 21 | 22 | 22 | 22 | 22 |
| MA7 | 21 | 22 | 22 | 23 | 23 | 23 | 23 |
| MA8 | 22 | 23 | 23 | 24 | 24 | 24 | 24 |
| MA9 | 23 | 24 | 24 | 25 | 25 | 25 | 25 |
| MA10 | 24 | 25 | 25 | 26 | 26 | 26 | 26 |
| MA11 | 25 | 26 | 26 | 27 | 27 | 27 | 27 |
| MA12 | 26 | 27 | 27 | - | 28 | 28 | 28 |
| MA13 | - | - | 28 | - | - | 29 | 29 |
| MA14 | - | - | - | - | - | - | 30 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-27. MA Table (AMTSEL = 2'b01) in 8-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MA1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MA2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA9 | - | 9 | 9 | 9 | 9 | 9 | 9 |
| MA10 | - | - | - | - | - | | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 22 | 23 | 24 | 22 | 23 | 24 | 25 |
| BA1 | 23 | 24 | 25 | 23 | 24 | 25 | 26 |
| BA2 | - | - | - | 24 | 25 | 26 | 27 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 9 | 10 | 10 | 10 | 10 | 10 | 10 |
| MA1 | 10 | 11 | 11 | 11 | 11 | 11 | 11 |
| MA2 | 11 | 12 | 12 | 12 | 12 | 12 | 12 |
| MA3 | 12 | 13 | 13 | 13 | 13 | 13 | 13 |
| MA4 | 13 | 14 | 14 | 14 | 14 | 14 | 14 |
| MA5 | 14 | 15 | 15 | 15 | 15 | 15 | 15 |
| MA6 | 15 | 16 | 16 | 16 | 16 | 16 | 16 |
| MA7 | 16 | 17 | 17 | 17 | 17 | 17 | 17 |
| MA8 | 17 | 18 | 18 | 18 | 18 | 18 | 18 |
| MA9 | 18 | 19 | 19 | 19 | 19 | 19 | 19 |
| MA10 | 19 | 20 | 20 | 20 | 20 | 20 | 20 |
| MA11 | 20 | 21 | 21 | 21 | 21 | 21 | 21 |
| MA12 | 21 | 22 | 22 | - | 22 | 22 | 22 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA13 | - | - | 23 | - | - | 23 | 23 |
| MA14 | - | - | - | - | - | - | 24 |
| MA15 | - | - | - | - | | - | - |

**Table 9-28. MA Table (AMTSEL = 2'b01) in 16-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MA1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA8 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| MA9 | - | 10 | 10 | 10 | 10 | 10 | 10 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 23 | 24 | 25 | 23 | 24 | 25 | 26 |
| BA1 | 24 | 25 | 26 | 24 | 25 | 26 | 27 |
| BA2 | - | - | - | 25 | 26 | 27 | 28 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 10 | 11 | 11 | 11 | 11 | 11 | 11 |
| MA1 | 11 | 12 | 12 | 12 | 12 | 12 | 12 |
| MA2 | 12 | 13 | 13 | 13 | 13 | 13 | 13 |
| MA3 | 13 | 14 | 14 | 14 | 14 | 14 | 14 |
| MA4 | 14 | 15 | 15 | 15 | 15 | 15 | 15 |
| MA5 | 15 | 16 | 16 | 16 | 16 | 16 | 16 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA6 | 16 | 17 | 17 | 17 | 17 | 17 | 17 |
| MA7 | 17 | 18 | 18 | 18 | 18 | 18 | 18 |
| MA8 | 18 | 19 | 19 | 19 | 19 | 19 | 19 |
| MA9 | 19 | 20 | 20 | 20 | 20 | 20 | 20 |
| MA10 | 20 | 21 | 21 | 21 | 21 | 21 | 21 |
| MA11 | 21 | 22 | 22 | 22 | 22 | 22 | 22 |
| MA12 | 22 | 23 | 23 | - | 23 | 23 | 23 |
| MA13 | - | - | 24 | - | - | 24 | 24 |
| MA14 | - | - | - | - | - | - | 25 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-29. MA Table (AMTSEL = 2'b01) in 32-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA5 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA6 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA7 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| MA8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| MA9 | - | 11 | 11 | 11 | 11 | 11 | 11 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 24 | 25 | 26 | 24 | 25 | 26 | 27 |
| BA1 | 25 | 26 | 27 | 25 | 26 | 27 | 28 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| BA2 | - | - | - | 26 | 27 | 28 | 29 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 11 | 12 | 12 | 12 | 12 | 12 | 12 |
| MA1 | 12 | 13 | 13 | 13 | 13 | 13 | 13 |
| MA2 | 13 | 14 | 14 | 14 | 14 | 14 | 14 |
| MA3 | 14 | 15 | 15 | 15 | 15 | 15 | 15 |
| MA4 | 15 | 16 | 16 | 16 | 16 | 16 | 16 |
| MA5 | 16 | 17 | 17 | 17 | 17 | 17 | 17 |
| MA6 | 17 | 18 | 18 | 18 | 18 | 18 | 18 |
| MA7 | 18 | 19 | 19 | 19 | 19 | 19 | 19 |
| MA8 | 19 | 20 | 20 | 20 | 20 | 20 | 20 |
| MA9 | 20 | 21 | 21 | 21 | 21 | 21 | 21 |
| MA10 | 21 | 22 | 22 | 22 | 22 | 22 | 22 |
| MA11 | 22 | 23 | 23 | 23 | 23 | 23 | 23 |
| MA12 | 23 | 24 | 24 | - | 24 | 24 | 24 |
| MA13 | - | - | 25 | - | - | 25 | 25 |
| MA14 | - | - | - | - | - | - | 26 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-30. MA Table (AMTSEL = 2'b01) in 64-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA3 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA4 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA5 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| MA6 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA7 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| MA8 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| MA9 | - | 12 | 12 | 12 | 12 | 12 | 12 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 25 | 26 | 27 | 25 | 26 | 27 | 28 |
| BA1 | 26 | 27 | 28 | 26 | 27 | 28 | 29 |
| BA2 | - | - | - | 27 | 28 | 29 | 30 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 12 | 13 | 13 | 13 | 13 | 13 | 13 |
| MA1 | 13 | 14 | 14 | 14 | 14 | 14 | 14 |
| MA2 | 14 | 15 | 15 | 15 | 15 | 15 | 15 |
| MA3 | 15 | 16 | 16 | 16 | 16 | 16 | 16 |
| MA4 | 16 | 17 | 17 | 17 | 17 | 17 | 17 |
| MA5 | 17 | 18 | 18 | 18 | 18 | 18 | 18 |
| MA6 | 18 | 19 | 19 | 19 | 19 | 19 | 19 |
| MA7 | 19 | 20 | 20 | 20 | 20 | 20 | 20 |
| MA8 | 20 | 21 | 21 | 21 | 21 | 21 | 21 |
| MA9 | 21 | 22 | 22 | 22 | 22 | 22 | 22 |
| MA10 | 22 | 23 | 23 | 23 | 23 | 23 | 23 |
| MA11 | 23 | 24 | 24 | 24 | 24 | 24 | 24 |
| MA12 | 24 | 25 | 25 | - | 25 | 25 | 25 |
| MA13 | - | - | 26 | - | - | 26 | 26 |
| MA14 | - | - | - | - | - | - | 27 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-31. MA Table (AMTSEL = 2'b10) in 16-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MA1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA5 | 8 | 8 | 8 | 9 | 9 | 9 | 9 |
| MA6 | 9 | 9 | 9 | 10 | 10 | 10 | 10 |
| MA7 | 10 | 10 | 10 | 11 | 11 | 11 | 11 |
| MA8 | 11 | 11 | 11 | 12 | 12 | 12 | 12 |
| MA9 | - | 12 | 12 | 13 | 13 | 13 | 13 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| BA1 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| BA2 | - | - | - | 8 | 8 | 8 | 8 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 12 | 13 | 13 | 14 | 14 | 14 | 14 |
| MA1 | 13 | 14 | 14 | 15 | 15 | 15 | 15 |
| MA2 | 14 | 15 | 15 | 16 | 16 | 16 | 16 |
| MA3 | 15 | 16 | 16 | 17 | 17 | 17 | 17 |
| MA4 | 16 | 17 | 17 | 18 | 18 | 18 | 18 |
| MA5 | 17 | 18 | 18 | 19 | 19 | 19 | 19 |
| MA6 | 18 | 19 | 19 | 20 | 20 | 20 | 20 |
| MA7 | 19 | 20 | 20 | 21 | 21 | 21 | 21 |
| MA8 | 20 | 21 | 21 | 22 | 22 | 22 | 22 |
| MA9 | 21 | 22 | 22 | 23 | 23 | 23 | 23 |
| MA10 | 22 | 23 | 23 | 24 | 24 | 24 | 24 |
| MA11 | 23 | 24 | 24 | 25 | 25 | 25 | 25 |
| MA12 | 24 | 25 | 25 | - | 26 | 26 | 26 |
| MA13 | - | - | 26 | - | - | 27 | 27 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA14 | - | - | - | - | - | - | 28 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-32. MA Table (AMTSEL = 2'b10) in 32-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| MA1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA4 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA5 | 9 | 9 | 9 | 10 | 10 | 10 | 10 |
| MA6 | 10 | 10 | 10 | 11 | 11 | 11 | 11 |
| MA7 | 11 | 11 | 11 | 12 | 12 | 12 | 12 |
| MA8 | 12 | 12 | 12 | 13 | 13 | 13 | 13 |
| MA9 | - | 13 | 13 | 14 | 14 | 14 | 14 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| BA1 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| BA2 | - | - | - | 9 | 9 | 9 | 9 |
| **Row** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** | **RA** |
| MA0 | 13 | 14 | 14 | 15 | 15 | 15 | 15 |
| MA1 | 14 | 15 | 15 | 16 | 16 | 16 | 16 |
| MA2 | 15 | 16 | 16 | 17 | 17 | 17 | 17 |
| MA3 | 16 | 17 | 17 | 18 | 18 | 18 | 18 |
| MA4 | 17 | 18 | 18 | 19 | 19 | 19 | 19 |
| MA5 | 18 | 19 | 19 | 20 | 20 | 20 | 20 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| MA6 | 19 | 20 | 20 | 21 | 21 | 21 | 21 |
| MA7 | 20 | 21 | 21 | 22 | 22 | 22 | 22 |
| MA8 | 21 | 22 | 22 | 23 | 23 | 23 | 23 |
| MA9 | 22 | 23 | 23 | 24 | 24 | 24 | 24 |
| MA10 | 23 | 24 | 24 | 25 | 25 | 25 | 25 |
| MA11 | 24 | 25 | 25 | 26 | 26 | 26 | 26 |
| MA12 | 25 | 26 | 26 | - | 27 | 27 | 27 |
| MA13 | - | - | 27 | - | - | 28 | 28 |
| MA14 | - | - | - | - | - | - | 29 |
| MA15 | - | - | - | - | - | - | - |

**Table 9-33. MA Table (AMTSEL = 2'b10) in 64-bit Mode**

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| **TYPE** | **13X9X2** | **13X10X2** | **14X10X2** | **12X10X3** | **13X10X3** | **14X10X3** | **15X10X3** |
| **SELECT** | **3'b000** | **3'b001** | **3'b010** | **3'b011** | **3'b100** | **3'b101** | **3'b110** |
| **Column** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** | **CA** |
| MA0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| MA1 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MA2 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| MA3 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| MA4 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| MA5 | 10 | 10 | 10 | 11 | 11 | 11 | 11 |
| MA6 | 11 | 11 | 11 | 12 | 12 | 12 | 12 |
| MA7 | 12 | 12 | 12 | 13 | 13 | 13 | 13 |
| MA8 | 13 | 13 | 13 | 14 | 14 | 14 | 14 |
| MA9 | - | 14 | 14 | 15 | 15 | 15 | 15 |
| MA10 | - | - | - | - | - | - | - |
| **Bank** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** | **BA** |
| BA0 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| BA1 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| BA2 | - | - | - | 10 | 10 | 10 | 10 |

| MATABLE | Row Address x Column Address x Bank Address | | | | | | |
|---|---|---|---|---|---|---|---|
| TYPE | 13X9X2 | 13X10X2 | 14X10X2 | 12X10X3 | 13X10X3 | 14X10X3 | 15X10X3 |
| SELECT | 3'b000 | 3'b001 | 3'b010 | 3'b011 | 3'b100 | 3'b101 | 3'b110 |
| Row | RA | RA | RA | RA | RA | RA | RA |
| MA0 | 14 | 15 | 15 | 16 | 16 | 16 | 16 |
| MA1 | 15 | 16 | 16 | 17 | 17 | 17 | 17 |
| MA2 | 16 | 17 | 17 | 18 | 18 | 18 | 18 |
| MA3 | 17 | 18 | 18 | 19 | 19 | 19 | 19 |
| MA4 | 18 | 19 | 19 | 20 | 20 | 20 | 20 |
| MA5 | 19 | 20 | 20 | 21 | 21 | 21 | 21 |
| MA6 | 20 | 21 | 21 | 22 | 22 | 22 | 22 |
| MA7 | 21 | 22 | 22 | 23 | 23 | 23 | 23 |
| MA8 | 22 | 23 | 23 | 24 | 24 | 24 | 24 |
| MA9 | 23 | 24 | 24 | 25 | 25 | 25 | 25 |
| MA10 | 24 | 25 | 25 | 26 | 26 | 26 | 26 |
| MA11 | 25 | 26 | 26 | 27 | 27 | 27 | 27 |
| MA12 | 26 | 27 | 27 | - | 28 | 28 | 28 |
| MA13 | - | - | 28 | - | - | 29 | 29 |
| MA14 | - | - | - | - | - | - | 30 |
| MA15 | - | - | - | - | - | - | - |

# Chapter 10

# DMA Controller

This chapter contains the following sections:


- 10.1 General Description
- 10.2 Features
- 10.3 Block Diagram
- 10.4 Programming Model
- 10.5 Programming Sequence

## 10.1 General Description

The Direct Memory Access (DMA) controller is designed to enhance the system performance and reduce the processor-interrupt generation. The system efficiency can be improved by employing the high-speed data transfers between the system and the device. The DMA controller provides up to eight configurable channels for the memory-to-memory, memory-to-peripheral, peripheral-to-peripheral, and peripheral-to-memory transfers with a shared buffer.

## 10.2 Features

- An AHB slave port for DMA controller configuration
- Two AHB master interfaces for data transfers
- Up to eight configurable DMA channels
- 16 DMA requests/acknowledges
- Supports chain transfer
- Supports memory-to-memory, memory-to-peripheral, peripheral-to-peripheral, and peripheral-to-memory transfers
- Group round-robin arbitration scheme with four priority levels
- Supports the hardware handshake
- Supports big-endian and little-endian

## 10.3 Block Diagram

Figure 10-1 illustrates the functional block diagram of the DMA controller without the simple bridge configuration.

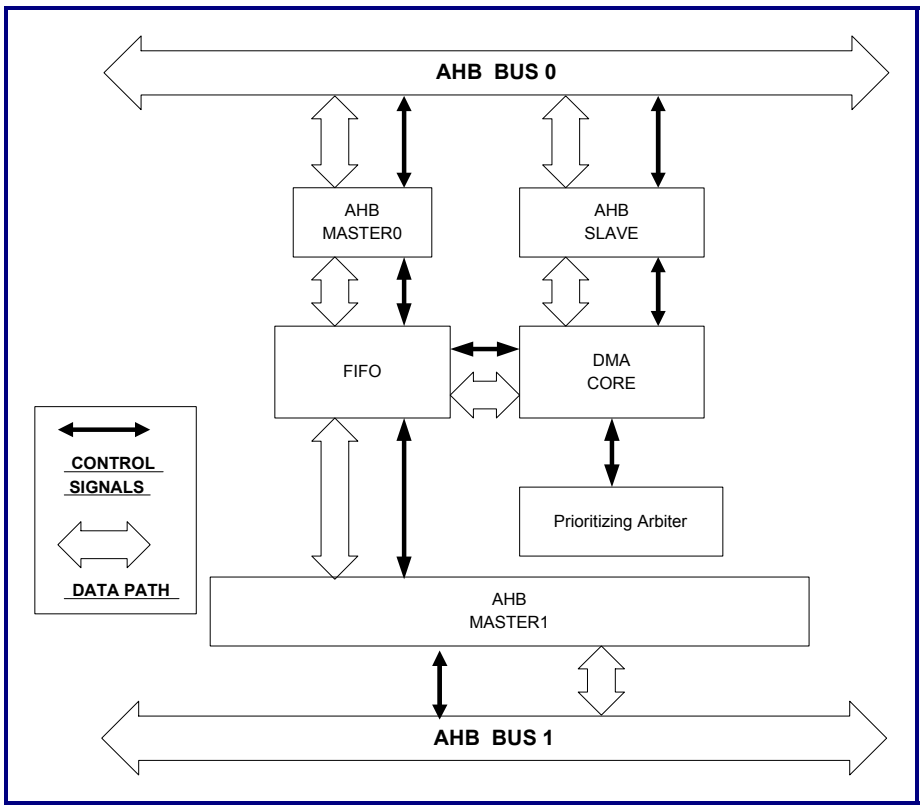



**Figure 10-1. Functional Block Diagram of DMA Controller Without Simple Bridge Configuration**

The DMA controller consists of five main building blocks: Two AHB master interfaces, one AHB slave interface, one FIFO buffer, and one DMA core.

### 10.3.1 AHB Master Interface

The DMA controller has two AHB master interfaces. The AHB Bus 0 master is necessary, whereas the AHB Bus 1 master is optional through the configuration setting. They can transfer data between the system and DMA FIFO.

### 10.3.2 AHB Slave Interface

The DMA controller has a slave interface. The system can configure the DMA controller or access the devices on AHB Bus 1 (If the simple bridge configuration option is turned on under the DMAC32 condition) through this AHB slave interface.

### 10.3.3 FIFO Buffer

The FIFO buffer provides the buffer between the source and the destination.

### 10.3.4 DMA Core

The DMA core is configurable up to an 8-channel DMA engine. It supports data transfers between the two AHB interfaces. Both the source and destination can be on either AHB Bus 0 or AHB Bus 1. Each channel can be assigned with a group priority level, and the same group priority are serviced in the round-robin fashion.

## 10.4 Programming Model

### 10.4.1 Prioritizing Arbiter

The DMA controller uses the 4-group priority and the round-robin scheme to select which channel to serve. The arbitration is based on the priority level of the channels. If the channels have the same priority level, the arbitration will then be based on the round robin scheme. Each channel has a 2-bit priority value associated with it. A value of 3 indicates the highest priority level and a value of 0 indicates the lowest priority. The arbitration scheme is shown in Figure 10-2 on the next page.

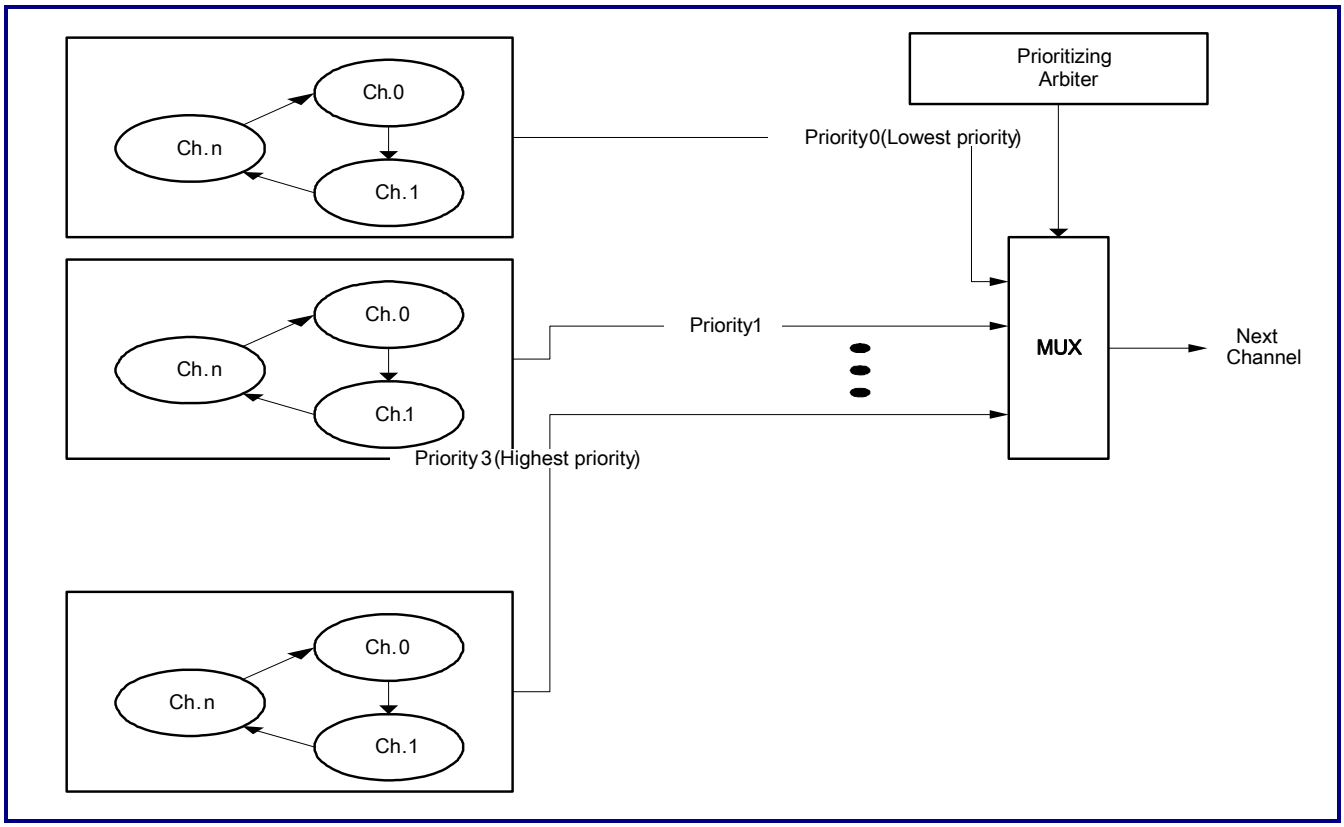



**Figure 10-2. Arbitration Scheme**

### 10.4.2 Chain Transfer

The DMA controller provides the chain transfer function. The multiple blocks can be transferred consecutively without processing the CPU after the end of transfer of each block. For each channel, before starting the chain transfer, a Linked List structure (Shown in Figure 10-3) must be built in some memory area in the system. The Linked List descriptor pointer register (**Cn_LLP**) for this channel must be programmed to point to the head of this linked list structure.

For each channel **n**, after finishing the first block data transfer, the DMA engine will check whether the linked list descriptor pointer address (Bits[31:2]) of the Linked List descriptor pointer register (**Cn_LLP**) is 0 or not. If the value is a non-zero, it means that the chain transfer function is enabled for this channel. In such a case, the DMA controller will fetch the first channel of the Linked List descriptor from the memory attached to the AHB bus 0 or AHB bus 1 depending on bit 0 of the Linked List descriptor pointer register (**Cn_LLP**). After finishing the data transfer of the first channel of the Linked List descriptor, the DMA engine will continue the data transfer of the next channel until the end of the Linked List structure. As

shown in Figure 10-3, there are three Linked List descriptors in the Linked List structure. Therefore, there are four consecutive blocks to be transferred without CPU processing. The first block is specified in **Cn_CSR** and **Cn_SIZE**. The second block is specified in the first Linked List descriptor. The third block is specified in the second Linked List descriptor. The fourth block is specified in the third Linked List descriptor. Please note that for the first block data transfer, the source, destination, total size, and control are based on **Cn_SrcAddr**, **Cn_DstAddr**, **Cn_Size**, and **Cn_CSR**. For transfer of the second and later blocks, the source, destination, total size, and control are based on the Linked List descriptor.

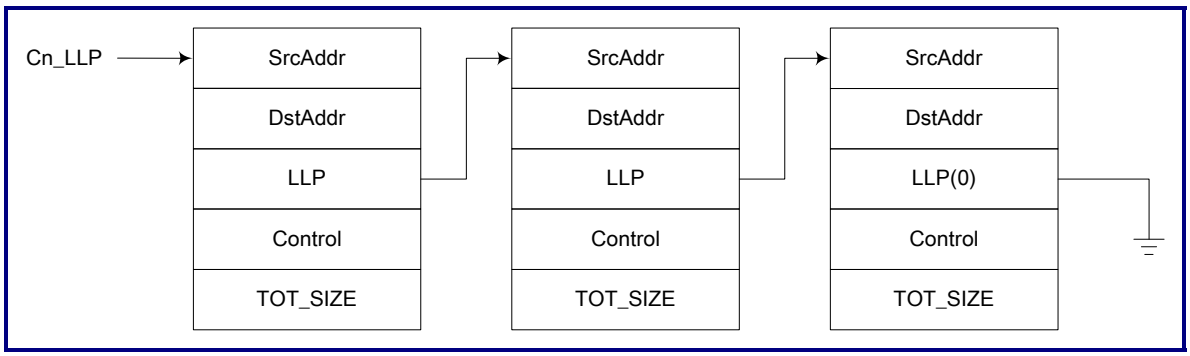



**Figure 10-3. Linked List Structure of Chain Transfer Operation**

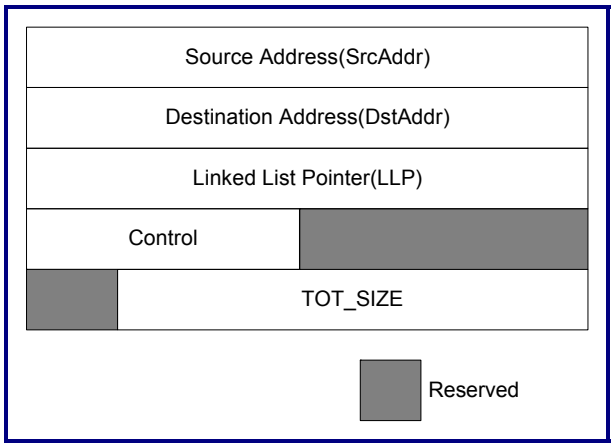



**Figure 10-4. Linked List Descriptor**

Each time the DMA controller fetches the linked list descriptor (Word-aligned), SrcAddr is copied to the **Cn_SrcAddr** register, DstAddr is copied to the **Cn_DstAddr** register, LLP is copied to the **Cn_LLP** register, Total Transfer Size is copied to the TOT_SIZE field in the **Cn_SIZE** register, and Control is copied to the respective bits in the **Cn_CSR** register.

**Table 10-1. Address Map for Linked List Descriptor (Base Address: Cn_LLP[31:2])**

| Name | Offset | Width | Description |
|---|---|---|---|
| SrcAddr | +0 | 32 | Source address |
| DstAddr | +4 | 32 | Destination address |
| LLP | +8 | 32 | Linked list pointer |
| Control | +C | 32 | Control and total transfer size (Please refer to Table 10-2) |
| Total size | +10 | 32 | Total transfer size (Please refer to Table 10-3) |

**Table 10-2. Control Field Definition in Linked List Descriptor**

| Bit | Description |
|---|---|
| [31:29] | DMA_FF_TH: DMA FIFO threshold value (Same as DMA_FF_TH in **Cn_CSR**) |
| 28 | TC_MSK: Channel terminal count status mask (Same as TC_MSK in **Cn_CSR**) |
| [27:25] | SRC_WIDTH: Source transfer width (Same as SRC_WIDTH in **Cn_CSR**) |
| [24:22] | DST_WIDTH: Destination transfer width (Same as DST_WIDTH in **Cn_CSR**) |
| [21:20] | SRCAD_CTL: Source address control (Same as SRCAD_CTL in **Cn_CSR**) |
| [19:18] | DSTAD_CTL: Destination address control (Same as DSTAD_CTL in **Cn_CSR**) |
| 17 | SRC_SEL: Source selection (Same as SRC_SEL in **Cn_CSR**) |
| 16 | DST_SEL: Destination selection (Same as DST_SEL in **Cn_CSR**) |
| [15:0] | Reserved |

**Table 10-3. Total Transfer Size Definition in Linked List Descriptor**

| Bit | Description |
|---|---|
| [21:0] | TOT_SIZE: Total transfer size (Same as TOT_SIZE in **Cn_SIZE**) |

For each channel, when the chain transfer is enabled, after finishing transfer of one block, the DMA engine will assert the interrupt (**dmaint**) to inform CPU that the data transfer is done. CPU should then de-assert the interrupt in time by writing '1' to the respective bit of the Terminal Count Interrupt Status Clear Register before completing the data transfer of the next block. However, in some cases, the CPU will not de-assert the interrupt in time before completing the data transfer of the next block. In such cases, CPU will not receive the completion of data transfer of two blocks. This will cause trouble to the CPU during the chain transfer operation.

For example, Figure 10-5 shows a chain transfer operation composed of transactions 1, 2, and 3, i.e., two linked list descriptors are fetched. At time $t_1$, transaction 1 is finished and DMA sets Cn_INT to assert **dmaint**. At time $t_2$, the CPU reads Cn_INT to know the interrupt source in the interrupt service routine. At time $t_3$, transaction 2 is finished and then DMA sets Cn_INT. At time $t_4$, the interrupt service routine for time $t_2$ clears Cn_INT. At time $t_5$, transaction 3 is finished, and the CPU does not know that transaction 2 had been finished (At time $t_4$, the software clears Cn_INT of transactions 1 and 2) and it causes an interrupt conflict. To avoid this conflict situation, DMA is designed to provide a LLP counter to record the transaction numbers when the chain transfer is enabled.

The LLP counter (Denoted as LLP_CNT, bits[19:16] of Channel Configuration Register) operation is shown in Figure 10-6. When ch_en is set to '1' (When the chain transfer is enabled), LLP_CNT is reset to '0'. When each transaction is finished, LLP_CNT is increased by 1. Please note that when the last transaction is finished, LLP_CNT is also increased by 1. When an error or abort condition happens, the chain transfer is stopped and ch_en is cleared to 0, and then LLP_CNT will also be increased by 1.

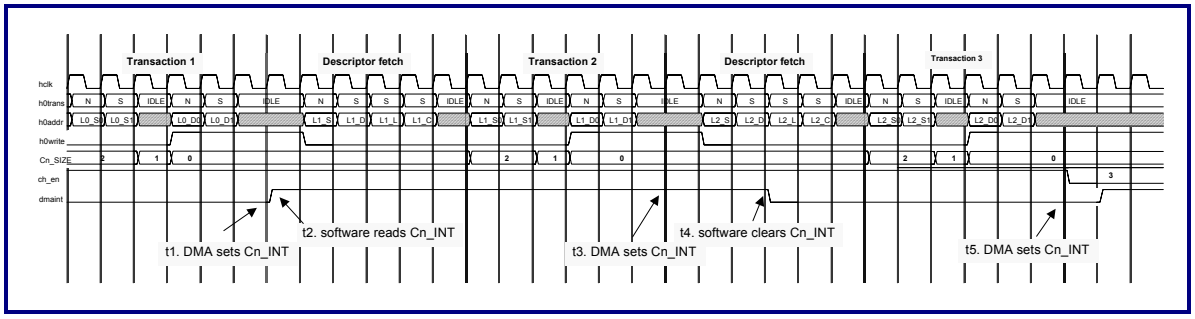



**Figure 10-5. Example of Interrupt Conflict During Chain Transfer Operation**

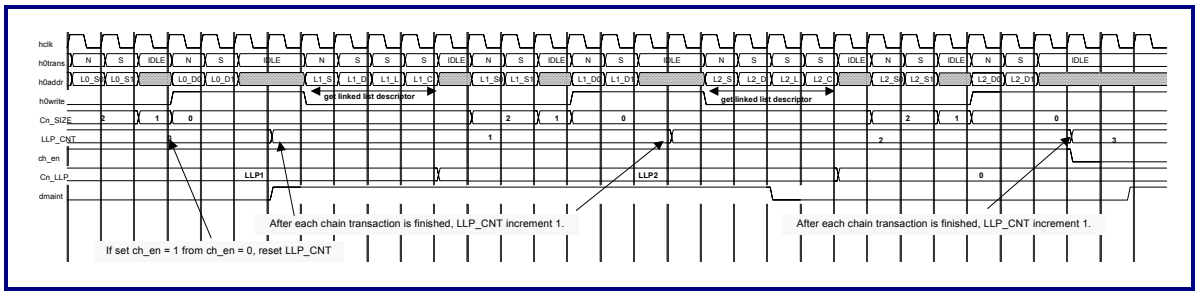



**Figure 10-6. LLP_CNT Operation**

### 10.4.3 DMA Hardware Handshake Mode

Each channel of the DMA controller must be programmed into either the DMA hardware handshake mode or DMA normal mode. The DMA hardware handshake mode will be explained in this section and the DMA normal mode will be explained in the next section.

The DMA hardware handshake mode can be enabled by setting bit 7 of Channel Control Register (**Cn_CSR**). For example, the DMA hardware handshake mode of channel 0 can be enabled by setting bit[7] of **C0_CSR**. If the DMA hardware handshake mode of channel 0 is enabled, after channel 0 wins the arbitration, the DMA controller will wait for the external DMA request to be asserted before starting the DMA transfer. Each time the DMA request is asserted, the controller transfers units equal to SRC_BURST_SIZE[1]. When the SRC_BURST_SIZE transfer is completed, the DMA controller asserts acknowledge and then re-arbitrates all DMA requests. After detecting the assertion of acknowledge, the external device should de-assert the DMA request to let the DMA controller de-assert acknowledge. Please note that DMA_request can not be asserted again until acknowledge is de-asserted. After the TOT_SIZE transfers complete, the DMA controller asserts TC[0] (Bit 0 of Terminal Count Status Register (TC)), **dma_tc[0],** and both the **dmaint_tc** and **dmaint** interrupts (If not masked). Figure 10-7 illustrates an example of the DMA transfer between two AHB interfaces in the hardware handshake mode. Figure 10-8 shows the hardware handshake protocol. Please note that the DMA controller de-asserts **dma_ack** and **dma_tc** at the same time **dma_req** is deasserted.

During the transfer, if the source or destination slave returns an ERROR response, DMA will set the ERR bit (Bits[7:0] of Error/Abort Status Register (ERR_ABT)) and terminate the DMA transfer at once (Please note that if the source and destination slaves are on different AHB buses and the other destination or source slave returns a RETRY response at the same time, the DMA controller will finish the RETRY cycle before setting the ERR bit). Then, if INT_ERR_MSK (Bit 1 of the Cn_CFG register) is not set, the DMA controller asserts **dmaint_err** and **dmaint**. In such a case, the DMA controller will not assert acknowledge (**dma_ack**) and the device must deassert **dma_req**. For example, Figure 10-9 shows an ERROR response happens when the DMA controller reads address S6 of source slave, and a RETRY cycle happens when the DMA controller reads address D2 of destination slave; therefore, it forces the DMA controller to complete the RETRY cycle before asserting the **dmaint_err** and **dmaint** interrupts. In such a case, **dma_ack** is not asserted by the DMA controller.

[1] SRC_BURST_SIZE is set according to **SRC_SIZE** in the **Cn_CSR** register.

During the transfer, if the software sets the abort bit (Bit 15 of Channel Control Register (Cn_CSR)), after finishing SRC_BURST_SIZE transfers or TOT_SIZE transfers, the DMA controller will set the ABT bit (Bits[23:16] of Error/Abort status register (ERR_ABT)) and terminate the DMA transfer at once. (Please note that if the source and destination slaves return a RETRY response at the same time, the DMA controller will finish the RETRY cycle before setting the ABT bit.) Then, if INT_ABT_MSK (Bit 2 of the Cn_CFG register) is '0', the DMA controller asserts **dmaint**. In such a case, the DMA controller will not assert acknowledge (**dma_ack**) and the device must deassert **dma_req** . For example, Figure 10-10 shows the abort operation when this channel is active. At $t_1$, the internal abort state is set, and after finishing SRC_BURST_SIZE transfers or TOT_SIZE transfers (Time $t_2$) and completing the RETRY cycle (Time $t_3$), the DMA controller asserts the **dmaint** interrupt without asserting **dma_ack** at time $t_4$.

Figure 10-11 shows the abort operation when this channel is inactive. At time t1, the internal abort state is set. Because the channel is inactive, the DMA controller asserts the **dmaint** interrupt but does not assert the **dma_ack** at time $t_2$.

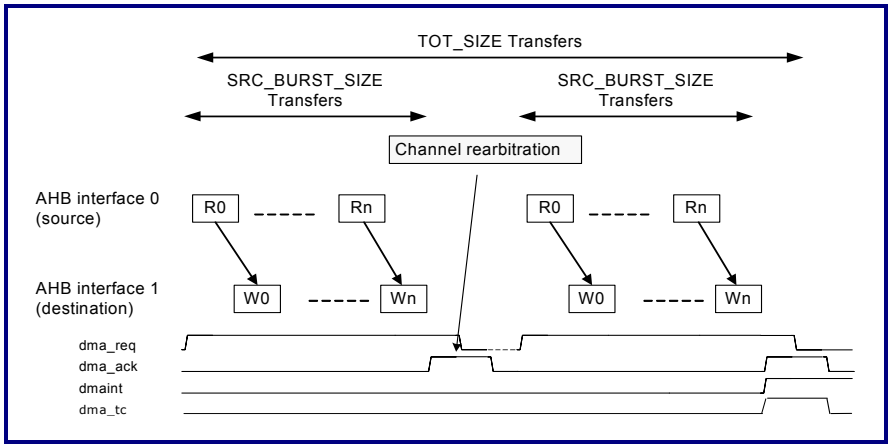



**Figure 10-7. Example of Hardware Handshake Mode Transfer**

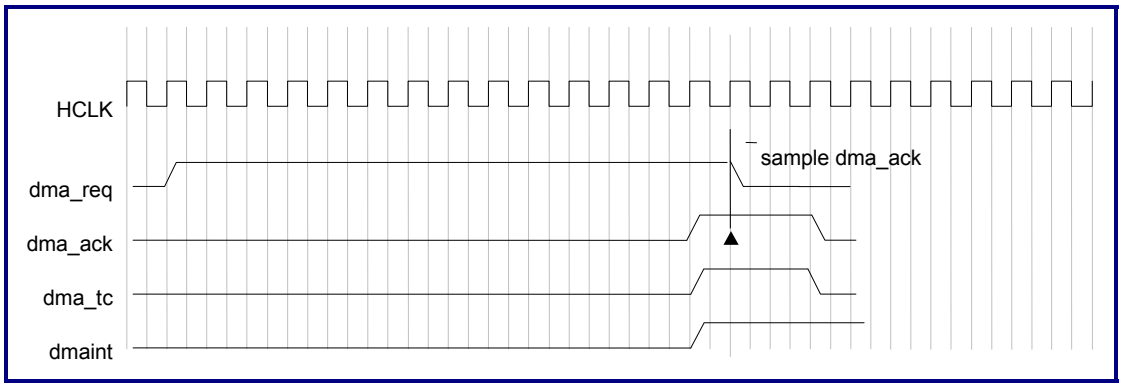



**Figure 10-8. DMA Hardware Handshake Protocol**

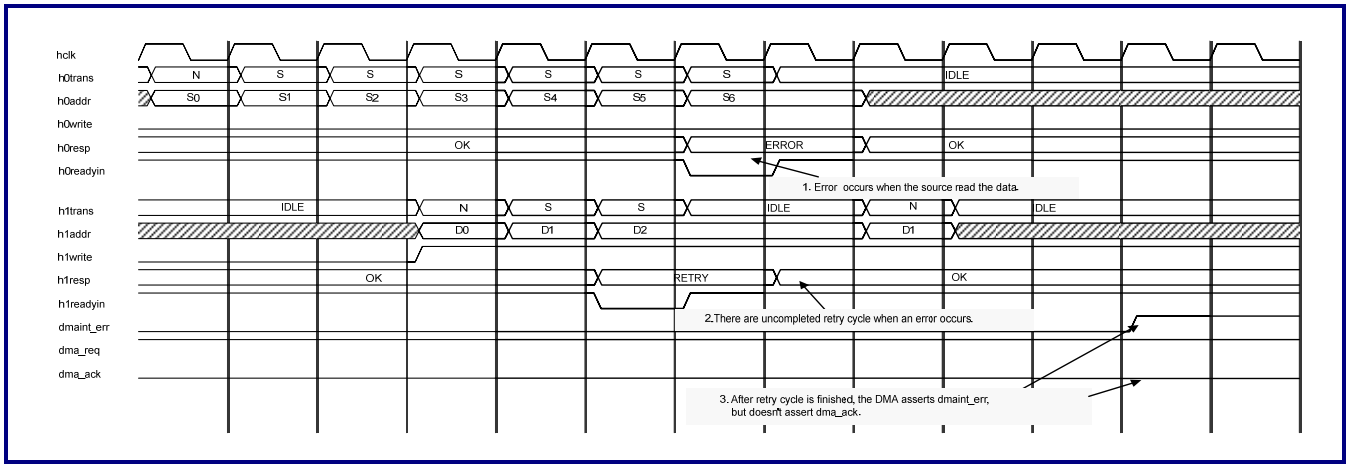



**Figure 10-9. ERROR Response Happens When in Hardware Handshake Mode Transfer**

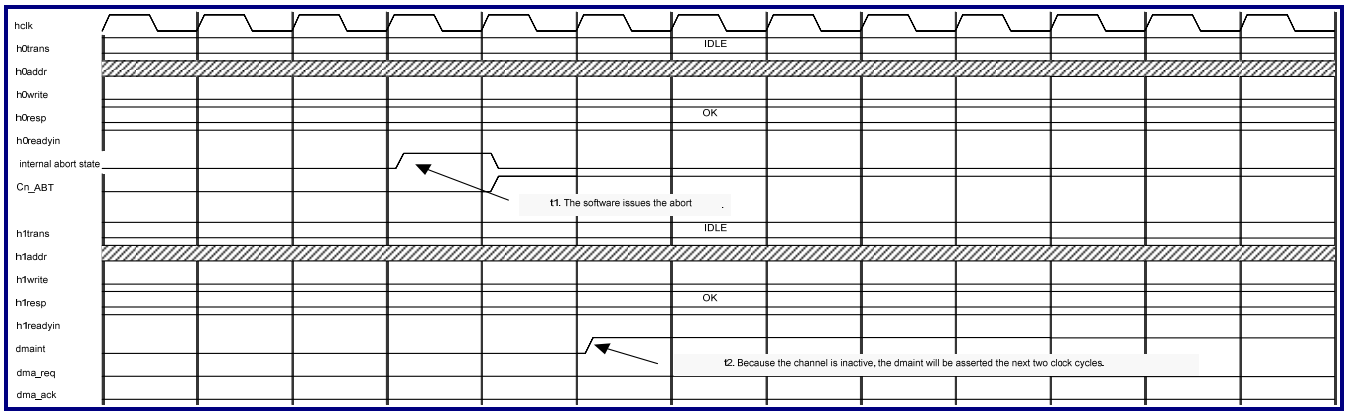



**Figure 10-10. Abort Operation When Channel is Active**

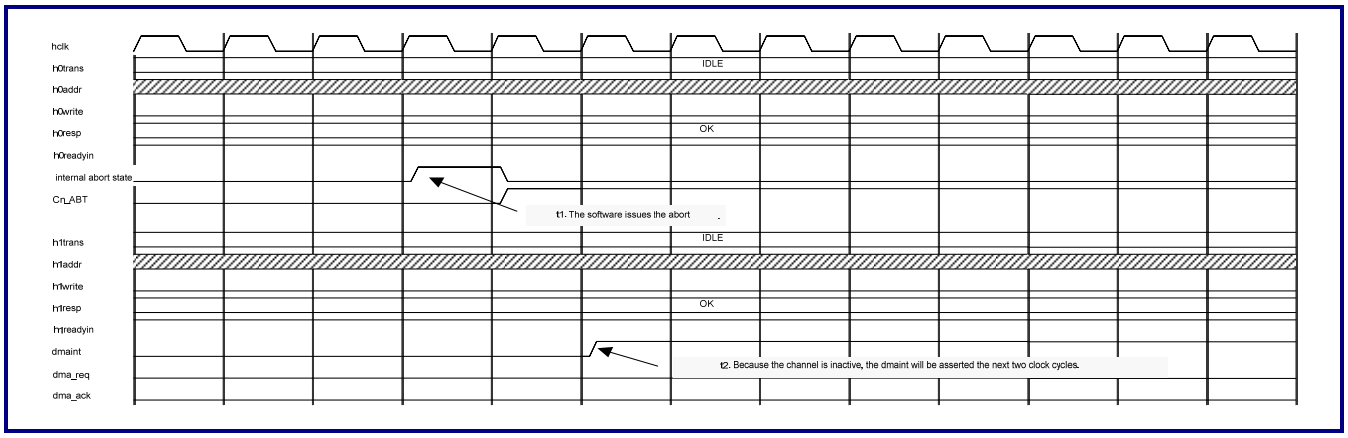



**Figure 10-11. Abort Operation When Channel is Inactive**

DMA provides 16 req/ack signal pairs. Every AHB device with the DMA access capability is associated with a specific req/ack pair. The following table lists the relationship of the AHB devices and the req/ack pairs.

**Table 10-4. AHB DMA REQ/ACK Mapping Table**

| DMA Request No. | AHB Device |
|---|---|
| 0 | SSP0 RX |
| 1 | SSP0 TX |
| 2 | SSP1 RX |
| 3 | SSP1 TX |
| 4 | SSP2 RX |
| 5 | SSP2 TX |
| 6 | SDC |
| 7 | Reserved |
| 8 | Reserved |
| 9 | NAND controller |
| 11 ~ 10 | Reserved |
| 12 | pwmtmr5 |
| 13 | pwmtmr6 |
| 14 | pwmtmr7 |
| 15 | pwmtmr8 |

### 10.4.4 DMA Normal Mode

In this mode, no external DMA request is needed. The DMA controller will automatically generate a transfer request signal internally. Figure 10-12 illustrates a transfer example for this mode. The error and abort operations are the same in the DMA hardware handshake mode.

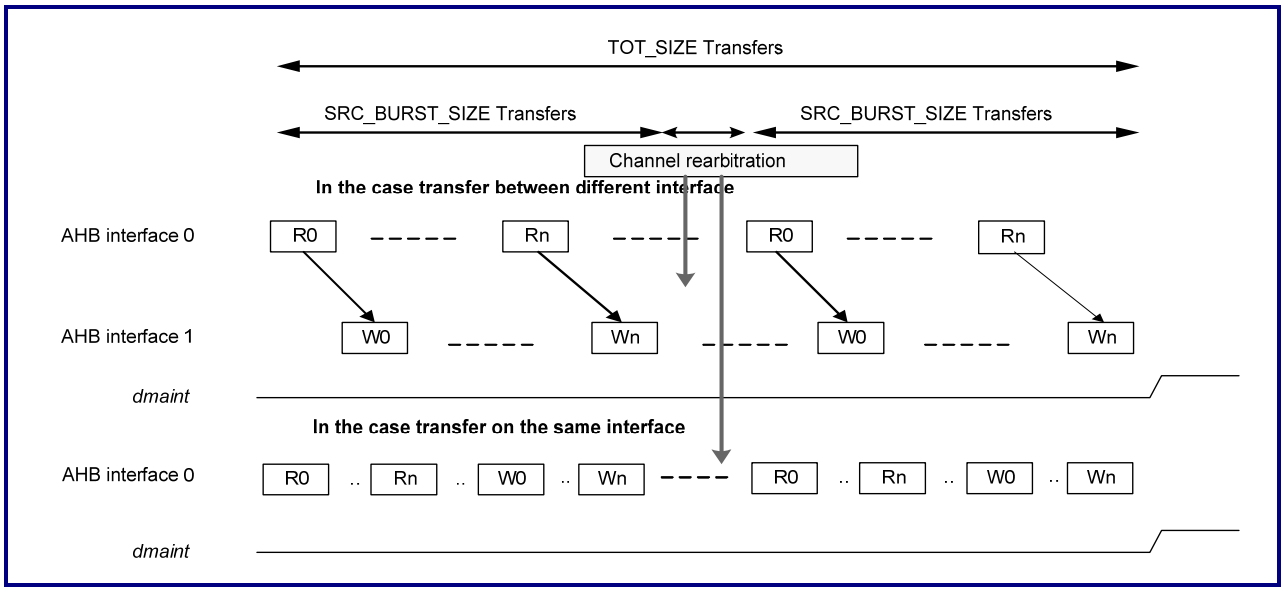



**Figure 10-12. Example of DMA Normal Mode Transfer**

### 10.4.5 Summary of DMA Controller Registers

This section describes all control and status registers in the DMA controller. The address column indicates the related address of the register in hexadecimal format, the width column indicates the number of bits of the register, and the access column indicates the access type of the register.

RW: Read and write access

RO: Read only

WO: Write 1 to clear

In addition, all reserved bits should always be written as 0. Reading these bits will return an undefined value.

**Table 10-5. Summary of DMA Controller Registers**

| Name | Addr. | Width | Access | Description |
|---|---|---|---|---|
| **Global registers** | | | | |
| INT | +0 | 8 | RO | Interrupt status register |
| INT_TC | +4 | 8 | RO | Interrupt for terminal count status register |
| INT_TC_CLR | +8 | 8 | WO | Interrupt for terminal count clear register |
| INT_ERR/ABT | +c | 32 | RO | Interrupt for Error/Abort status register |
| INT_ERR/ABT_CLR | +10 | 32 | WO | Interrupt for Error/Abort clear register |
| TC | +14 | 8 | RO | Terminal count status register |
| ERR/ABT | +18 | 32 | RO | Error/Abort status register |
| CH_EN | +1c | 8 | RO | Channel enable status register |
| CH_BUSY | +20 | 8 | RO | Channel busy register status register |
| CSR | +24 | 8 | RW | Main configuration status register |
| SYNC | +28 | 8 | RW | Sync register |
| **Channel 0 registers** | | | | |
| C0_CSR | +100 | 32 | RW | Channel 0 control register |
| C0_CFG | +104 | 32 | RW | Channel 0 configuration register |
| C0_SrcAddr | +108 | 32 | RW | Channel 0 source register |
| C0_DstAddr | +10c | 32 | RW | Channel 0 destination register |
| C0_LLP | +110 | 32 | RW | Channel 0 linked list pointer register |
| C0_SIZE | +114 | 32 | RW | Channel 0 transfer size register |
| **Channel 1 registers** | | | | |
| C1_CSR | +120 | 32 | RW | Channel 1 control register |
| C1_CFG | +124 | 32 | RW | Channel 1 configuration register |
| C1_SrcAddr | +128 | 32 | RW | Channel 1 source register |
| C1_DstAddr | +12c | 32 | RW | Channel 1 destination register |
| C1_LLP | +130 | 32 | RW | Channel 1 linked list pointer register |
| C1_SIZE | +134 | 32 | RW | Channel 1 transfer size register |
| **Channel 2 registers** | | | | |
| C2_CSR | +140 | 32 | RW | Channel 2 control register |
| C2_CFG | +144 | 32 | RW | Channel 2 configuration register |
| C2_SrcAddr | +148 | 32 | RW | Channel 2 source register |
| C2_DstAddr | +14c | 32 | RW | Channel 2 destination register |
| C2_LLP | +150 | 32 | RW | Channel 2 linked list pointer register |
| C2_SIZE | +154 | 32 | RW | Channel 2 transfer size register |

| Name | Addr. | Width | Access | Description |
|---|---|---|---|---|
| **Channel 3 registers** | | | | |
| C3_CSR | +160 | 32 | RW | Channel 3 control register |
| C3_CFG | +164 | 32 | RW | Channel 3 configuration register |
| C3_SrcAddr | +168 | 32 | RW | Channel 3 source register |
| C3_DstAddr | +16c | 32 | RW | Channel 3 destination register |
| C3_LLP | +170 | 32 | RW | Channel 3 linked list pointer register |
| C3_SIZE | +174 | 32 | RW | Channel 3 transfer size register |
| **Channel 4 registers** | | | | |
| C4_CSR | +180 | 32 | RW | Channel 4 control register |
| C4_CFG | +184 | 32 | RW | Channel 4 configuration register |
| C4_SrcAddr | +188 | 32 | RW | Channel 4 source register |
| C4_DstAddr | +18c | 32 | RW | Channel 4 destination register |
| C4_LLP | +190 | 32 | RW | Channel 4 linked list pointer register |
| C4_SIZE | +194 | 32 | RW | Channel 4 transfer size register |
| **Channel 5 registers** | | | | |
| C5_CSR | +1a0 | 32 | RW | Channel 5 control register |
| C5_CFG | +1a4 | 32 | RW | Channel 5 configuration register |
| C5_SrcAddr | +1a8 | 32 | RW | Channel 5 source register |
| C5_DstAddr | +1ac | 32 | RW | Channel 5 destination register |
| C5_LLP | +1b0 | 32 | RW | Channel 5 linked list pointer register |
| C5_SIZE | +1b4 | 32 | RW | Channel 5 transfer size register |
| **Channel 6 registers** | | | | |
| C6_CSR | +1c0 | 32 | RW | Channel 6 control register |
| C6_CFG | +1c4 | 32 | RW | Channel 6 configuration register |
| C6_SrcAddr | +1c8 | 32 | RW | Channel 6 source register |
| C6_DstAddr | +1cc | 32 | RW | Channel 6 destination register |
| C6_LLP | +1d0 | 32 | RW | Channel 6 linked list pointer register |
| C6_SIZE | +1d4 | 32 | RW | Channel 6 transfer size register |
| **Channel 7 registers** | | | | |
| C7_CSR | +1e0 | 32 | RW | Channel 7 control register |
| C7_CFG | +1e4 | 32 | RW | Channel 7 configuration register |
| C7_SrcAddr | +1e8 | 32 | RW | Channel 7 source register |
| C7_DstAddr | +1ec | 32 | RW | Channel 7 destination register |
| C7_LLP | +1f0 | 32 | RW | Channel 7 linked list pointer register |
| C7_SIZE | +1f4 | 32 | RW | Channel 7 transfer size register |

### 10.4.5.1 Interrupt Status Register (INT) (Offset = 0x00)

This register is used to keep the result of the following equation:

INT[***n***] = INT_ABT[***n***] | INT_ERR[***n***] | INT_TC[***n***]

where "***n***" is channel number ***n***

"|" is logic or.

INT_ABT[n] means bits[23:16] of the error/abort interrupt status register (INT_ERR/ABT). INT_ERR[n] means bits[7:0] of the error/abort interrupt status register (INT_ERR/ABT). INT_TC[n] means bits[7:0] of the terminal count interrupt status register (INT_TC). **dmaint** is asserted by the DMA controller if at least one bit of this register is set; on the other hand, **dmaint** is de-asserted by the DMA controller if no bit of this register is set. If the maximum channel number of DMA is configured as ***n***, bit 7 to bit ***n*** of this register will be reserved.

**Table 10-6. Interrupt Status Register (INT)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 7 | INT[7] | RO | The result of INT_ABT[7]\|INT_ERR[7] \| INT_TC[7]<br>0: Channel 7 has no pending interrupt.<br>1: Channel 7 has a pending interrupt. |
| 6 | INT[6] | RO | The result of INT_ABT[6]\|INT_ERR[6] \| INT_TC[6]<br>0: Channel 6 has no pending interrupt.<br>1: Channel 6 has a pending interrupt. |
| 5 | INT[5] | RO | The result of INT_ABT[5]\|INT_ERR[5] \| INT_TC[5]<br>0: Channel 5 has no pending interrupt.<br>1: Channel 5 has a pending interrupt. |
| 4 | INT[4] | RO | The result of INT_ABT[4]\|INT_ERR[4] \| INT_TC[4]<br>0: Channel 4 has no pending interrupt.<br>1: Channel 4 has a pending interrupt. |
| 3 | INT[3] | RO | The result of INT_ABT[3]\|INT_ERR[3] \| INT_TC[3]<br>0: Channel 3 has no pending interrupt.<br>1: Channel 3 has a pending interrupt. |
| 2 | INT[2] | RO | The result of INT_ABT[2]\|INT_ERR[2] \| INT_TC[2]<br>0: Channel 2 has no pending interrupt.<br>1: Channel 2 has a pending interrupt. |

| Bit | Name | Access | Description |
|---|---|---|---|
| 1 | INT[1] | RO | The result of INT_ABT[1]\|INT_ERR[1] \| INT_TC[1]<br>0: Channel 1 has no pending interrupt.<br>1: Channel 1 has a pending interrupt. |
| 0 | INT[0] | RO | The result of INT_ABT[0]\|INT_ERR[0] \| INT_TC[0]<br>0: Channel 0 has no pending interrupt.<br>1: Channel 0 has a pending interrupt. |

### 10.4.5.2 Terminal Count Interrupt Status Register (INT_TC) (Offset = 0x04)

This register shows the status of the DMA terminal count interrupts after masking. The mask bit of these interrupts is bit 0 (INT_TC_MSK) of the channel configuration register (Cn_CFG). If this mask bit is set, the content of this register is always 0 no matter there exists a pending DMA terminal count interrupt or not. Please refer to Section 10.4.3 and Section 10.4.4 for the detailed information on the generation of these interrupts.

**Table 10-7. Terminal Count Interrupt Status Register (INT_TC)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 7 | INT_TC[7] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 7 has no pending interrupt.<br>1: Channel 7 has a pending interrupt. |
| 6 | INT_TC[6] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 6 has no pending interrupt.<br>1: Channel 6 has a pending interrupt. |
| 5 | INT_TC[5] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 5 has no pending interrupt.<br>1: Channel 5 has a pending interrupt. |
| 4 | INT_TC[4] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 4 has no pending interrupt.<br>1: Channel 4 has a pending interrupt. |
| 3 | INT_TC[3] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 3 has no pending interrupt.<br>1: Channel 3 has a pending interrupt. |

| Bit | Name | Access | Description |
|---|---|---|---|
| 2 | INT_TC[2] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 2 has no pending interrupt.<br>1: Channel 2 has a pending interrupt. |
| 1 | INT_TC[1] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 1 has no pending interrupt.<br>1: Channel 1 has a pending interrupt. |
| 0 | INT_TC[0] | RO | Status of the DMA terminal count interrupts after masking<br>0: Channel 0 has no pending interrupt.<br>1: Channel 0 has a pending interrupt. |

Value after reset: 0000h

### 10.4.5.3 Terminal Count Interrupt Status Clear Register (INT_TC_CLR) (Offset = 0x08)

Writing '1' to bit ***n*** of this register clears both INT_TC[n] and TC[n]. INT_TC[n] means bits[7:0] of terminal count interrupt status register (INT_TC) and TC[n] means bits[7:0] of terminal count status register (INT).

**Table 10-8. Terminal Count Interrupt Status Clear Register (INT_TC_CLR)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 7 | INT_TC_CLR[7] | WO | Write 1 to clear the INT_TC[7] and TC[7] statuses |
| 6 | INT_TC_CLR[6] | WO | Write 1 to clear the INT_TC[6] and TC[6] statuses |
| 5 | INT_TC_CLR[5] | WO | Write 1 to clear the INT_TC[5] and TC[5] statuses |
| 4 | INT_TC_CLR[4] | WO | Write 1 to clear the INT_TC[4] and TC[4] statuses |
| 3 | INT_TC_CLR[3] | WO | Write 1 to clear the INT_TC[3] and TC[3] statuses |
| 2 | INT_TC_CLR[2] | WO | Write 1 to clear the INT_TC[2] and TC[2] statuses |
| 1 | INT_TC_CLR[1] | WO | Write 1 to clear the INT_TC[1] and TC[1] statuses |
| 0 | INT_TC_CLR[0] | WO | Write 1 to clear the INT_TC[0] and TC[0] statuses |

### 10.4.5.4 Error/Abort Interrupt Status Register (INT_ERR/ABT) (Offset = 0x0C)

INT_ERR is the status of the DMA error interrupts after masking. The mask bit of these interrupts is bit[1] (INT_ERR_MSK) of the channel configuration register (Cn_CFG). If this mask bit is set, the content of INT_ERR[n] of this register is always '0' no matter a pending DMA error interrupt is existed or not. If an AHB ERROR response happens during the DMA transfer, the DMA controller will stop the current DMA transfer and set ERR[n] (Bits[7:0] of error/abort status register (ERR/ABT)) to '1'. If INT_ERR_MSK is not set, the DMA controller will set INT_ERR[n] to '1' and assert both **dmaint_err** and **dmaint** interrupts. Please note that **dmaint_err** is asserted by the DMA controller if at least one bit of INT_ERR[7:0] of this register is set; on the other hand, **dmaint_err** is de-asserted by the DMA controller if no bit of INT_ERR[7:0] of this register is set.

INT_ABT is the status of the DMA abort interrupts after masking. The mask bit of these interrupts is bit 2 (INT_ABT_MSK) of the channel configuration register (Cn_CFG). If this mask bit is set, the content of INT_ABT[n] of this register is always '0' no matter there exists a pending DMA abort interrupt or not. If the ABT bit (Bit 15 of the channel control register (Cn_CSR)) is set, the DMA controller will stop the current DMA transfer and set ABT[*n*] (Bits[23:16] of the error/abort status register (ERR/ABT)) to '1'. If INT_ABT_MSK is not set, the DMA controller will set INT_ABT[*n*] to '1' and assert **dmaint** interrupt.

**Table 10-9. Error/Abort Interrupt Status Register (INT_ERR/ABT)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:24] | - | RO | Reserved |
| 23 | INT_ABT[7] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 7 has no pending interrupt.<br>1: Channel 7 has a pending interrupt. |
| 22 | INT_ABT[6] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 6 has no pending interrupt.<br>1: Channel 6 has a pending interrupt. |
| 21 | INT_ABT[5] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 5 has no pending interrupt.<br>1: Channel 5 has a pending interrupt. |
| 20 | INT_ABT[4] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 4 has no pending interrupt.<br>1: Channel 4 has a pending interrupt. |

| Bit | Name | Access | Description |
|---|---|---|---|
| 19 | INT_ABT[3] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 3 has no pending interrupt.<br>1: Channel 3 has a pending interrupt. |
| 18 | INT_ABT[2] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 2 has no pending interrupt.<br>1: Channel 2 has a pending interrupt. |
| 17 | INT_ABT[1] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 1 has no pending interrupt.<br>1: Channel 1 has a pending interrupt. |
| 16 | INT_ABT[0] | RO | Status of the DMA abort interrupts after masking<br>0: Channel 0 has no pending interrupt.<br>1: Channel 0 has a pending interrupt. |
| [15:8] | - | RO | Reserved |
| 7 | INT_ERR[7] | RO | Status of the DMA error interrupts after masking<br>0: Channel 7 has no pending interrupt.<br>1: Channel 7 has a pending interrupt. |
| 6 | INT_ERR[6] | RO | Status of the DMA error interrupts after masking<br>0: Channel 6 has no pending interrupt.<br>1: Channel 6 has a pending interrupt. |
| 5 | INT_ERR[5] | RO | Status of the DMA error interrupts after masking<br>0: Channel 5 has no pending interrupt.<br>1: Channel 5 has a pending interrupt. |
| 4 | INT_ERR[4] | RO | Status of the DMA error interrupts after masking<br>0: Channel 4 has no pending interrupt.<br>1: Channel 4 has a pending interrupt. |
| 3 | INT_ERR[3] | RO | Status of the DMA error interrupts after masking<br>0: Channel 3 has no pending interrupt.<br>1: Channel 3 has a pending interrupt. |
| 2 | INT_ERR[2] | RO | Status of the DMA error interrupts after masking<br>0: Channel 2 has no pending interrupt.<br>1: Channel 2 has a pending interrupt. |
| 1 | INT_ERR[1] | RO | Status of the DMA error interrupts after masking<br>0: Channel 1 has no pending interrupt.<br>1: Channel 1 has a pending interrupt. |

| Bit | Name | Access | Description |
|---|---|---|---|
| 0 | INT_ERR[0] | RO | Status of the DMA error interrupts after masking<br>0: Channel 0 has no pending interrupt.<br>1: Channel 0 has a pending interrupt. |

Value after reset: 0000h

### 10.4.5.5 Error/Abort Interrupt Status Clear Register (INT_ERR/ABT_CLR) (Offset = 0x10)

Writing '1' to bits[7:0] of this register clears both INT_ERR[n] and ERR[n]. Writing '1' to bits[23:16] of this register clears both INT_ABT[n] and ABT[n].

**Table 10-10. Error/Abort Interrupt Status Clear Register (INT_ERR/ABT_CLR)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:24] | - | WO | Reserved, write 0 |
| 23 | INT_ABT_CLR[7] | WO | Writing '1' to clear the INT_ABT[7] and ABT[7] status |
| 22 | INT_ABT_CLR[6] | WO | Writing '1' to clear the INT_ABT[6] and ABT[6] status |
| 21 | INT_ABT_CLR[5] | WO | Writing '1' to clear the INT_ABT[5] and ABT[5] status |
| 20 | INT_ABT_CLR[4] | WO | Writing '1' to clear the INT_ABT[4] and ABT[4] status |
| 19 | INT_ABT_CLR[3] | WO | Writing '1' to clear the INT_ABT[3] and ABT[3] status |
| 18 | INT_ABT_CLR[2] | WO | Writing '1' to clear the INT_ABT[2] and ABT[2] status |
| 17 | INT_ABT_CLR[1] | WO | Writing '1' to clear the INT_ABT[1] and ABT[1] status |
| 16 | INT_ABT_CLR[0] | WO | Writing '1' to clear the INT_ABT[0] and ABT[0] status |
| [15:8] | - | WO | Reserved, write 0 |
| 7 | INT_ERR_CLR[7] | WO | Writing '1' to clear the INT_ERR[7] and ERR[7] status |
| 6 | INT_ERR_CLR[6] | WO | Writing '1' to clear the INT_ERR[6] and ERR[6] status |
| 5 | INT_ERR_CLR[5] | WO | Writing '1' to clear the INT_ERR[5] and ERR[5] status |
| 4 | INT_ERR_CLR[4] | WO | Writing '1' to clear the INT_ERR[4] and ERR[4] status |
| 3 | INT_ERR_CLR[3] | WO | Writing '1' to clear the INT_ERR[3] and ERR[3] status |
| 2 | INT_ERR_CLR[2] | WO | Writing '1' to clear the INT_ERR[2] and ERR[2] status |
| 1 | INT_ERR_CLR[1] | WO | Writing '1' to clear the INT_ERR[1] and ERR[1] status |
| 0 | INT_ERR_CLR[0] | WO | Writing '1' to clear the INT_ERR[0] and ERR[0] status |

### 10.4.5.6 Terminal Count Status Register (TC) (Offset = 0x14)

This register shows the status of the DMA terminal count after masking. The mask bit for the DMA terminal count is bit 31 (TC_MSK) of the channel control register (Cn_CSR). If this mask bit of Cn_CSR is set, the TC[n] of this register is always 0 no matter a DMA terminal count happens or not. Please refer to Section 10.4.3 and Section 10.4.4 for more information on DMA terminal count.

**Table 10-11. Terminal Count Status Register (TC)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 7 | TC[7] | RO | Status of the DMA terminal count<br>0: Channel 7 has no terminal count status.<br>1: Channel 7 has a terminal count status. |
| 6 | TC[6] | RO | Status of the DMA terminal count<br>0: Channel 6 has no terminal count status.<br>1: Channel 6 has a terminal count status. |
| 5 | TC[5] | RO | Status of the DMA terminal count<br>0: Channel 5 has no terminal count status.<br>1: Channel 5 has a terminal count status. |
| 4 | TC[4] | RO | Status of the DMA terminal count<br>0: Channel 4 has no terminal count status.<br>1: Channel 4 has a terminal count status. |
| 3 | TC[3] | RO | Status of the DMA terminal count<br>0: Channel 3 has no terminal count status.<br>1: Channel 3 has a terminal count status. |
| 2 | TC[2] | RO | Status of the DMA terminal count<br>0: Channel 2 has no terminal count status.<br>1: Channel 2 has a terminal count status. |
| 1 | TC[1] | RO | Status of the DMA terminal count<br>0: Channel 1 has no terminal count status.<br>1: Channel 1 has a terminal count status. |
| 0 | TC[0] | RO | Status of the DMA terminal count<br>0: Channel 0 has no terminal count status.<br>1: Channel 0 has a terminal count status. |

Value after reset: 0000h

### 10.4.5.7 Error/Abort Status Register (ERR/ABT) (Offset = 0x18)

ERR is the status of the DMA error. If an AHB ERROR response happens during a DMA transfer, the DMA controller will stop the current DMA transfer and set ERR[n] to '1.' Then, if INT_ERR_MSK is not set, the DMA controller will set INT_ERR[n] to '1' and assert the **dmaint_err** and **dmaint** interrupts.

ABT is the status of the DMA abort. If the ABT bit (Bit 15 of the channel control register (Cn_CSR)) is set, the DMA controller will stop the current DMA transfer and set ABT[*n*] to '1'. If INT_ABT_MSK is not set, the DMA controller will set INT_ABT[*n*] to '1' and assert the **dmaint** interrupt.

**Table 10-12. Error/Abort Status Register (ERR/ABT)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:24] | - | RO | Reserved |
| 23 | ABT[7] | RO | Status of the DMA abort<br>0: Channel 7 has no abort status.<br>1: Channel 7 has an abort status. |
| 22 | ABT[6] | RO | Status of the DMA abort<br>0: Channel 6 has no abort status.<br>1: Channel 6 has an abort status. |
| 21 | ABT[5] | RO | Status of the DMA abort<br>0: Channel 5 has no abort status.<br>1: Channel 5 has an abort status. |
| 20 | ABT[4] | RO | Status of the DMA abort<br>0: Channel 4 has no abort status.<br>1: Channel 4 has an abort status. |
| 19 | ABT[3] | RO | Status of the DMA abort<br>0: Channel 3 has no abort status.<br>1: Channel 3 has an abort status. |
| 18 | ABT[2] | RO | Status of the DMA abort<br>0: Channel 2 has no abort status.<br>1: Channel 2 has an abort status. |
| 17 | ABT[1] | RO | Status of the DMA abort<br>0: Channel 1 has no abort status.<br>1: Channel 1 has an abort status. |

| Bit | Name | Access | Description |
|---|---|---|---|
| 16 | ABT[0] | RO | Status of the DMA abort<br>0: Channel 0 has no abort status.<br>1: Channel 0 has an abort status. |
| [15:8] | - | RO | Reserved |
| 7 | ERR[7] | RO | Status of the DMA error<br>0: Channel 7 has no error status.<br>1: Channel 7 has an error status. |
| 6 | ERR[6] | RO | Status of the DMA error<br>0: Channel 6 has no error status.<br>1: Channel 6 has an error status. |
| 5 | ERR[5] | RO | Status of the DMA error<br>0: Channel 5 has no error status.<br>1: Channel 5 has an error status. |
| 4 | ERR[4] | RO | Status of the DMA error<br>0: Channel 4 has no error status.<br>1: Channel 4 has an error status. |
| 3 | ERR[3] | RO | Status of the DMA error<br>0: Channel 3 has no error status.<br>1: Channel 3 has an error status. |
| 2 | ERR[2] | RO | Status of the DMA error<br>0: Channel 2 has no error status.<br>1: Channel 2 has an error status. |
| 1 | ERR[1] | RO | Status of the DMA error<br>0: Channel 1 has no error status.<br>1: Channel 1 has an error status. |
| 0 | ERR[0] | RO | Status of the DMA error<br>0: Channel 0 has no error status.<br>1: Channel 0 has an error status. |

Value after reset: 0000h

### 10.4.5.8 Channel Enable Status Register (CH_EN) (Offset = 0x1C)

This register shows the DMA channel enable status. It is a read-only register.

**Table 10-13. Channel Enable Status Register (CH_EN)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 7 | CH_EN[7] | RO | Status of the channel 7 CH_EN bit of the C7_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |
| 6 | CH_EN[6] | RO | Status of the channel 6 CH_EN bit of the C6_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |
| 5 | CH_EN[5] | RO | Status of the channel 5 CH_EN bit of the C5_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |
| 4 | CH_EN[4] | RO | Status of the channel 4 CH_EN bit of the C4_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |
| 3 | CH_EN[3] | RO | Status of the channel 3 CH_EN bit of the C3_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |
| 2 | CH_EN[2] | RO | Status of the channel 2 CH_EN bit of the C2_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |
| 1 | CH_EN[1] | RO | Status of the channel 1 CH_EN bit of the C1_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |
| 0 | CH_EN[0] | RO | Status of the channel 0 CH_EN bit of the C0_CSR register<br>0: CH_EN = 0<br>1: CH_EN = 1 |

Value after reset: 0000h

### 10.4.5.9 Channel Busy Status Register (CH_BUSY) (Offset = 0x20)

This register shows the DMA channel busy status. It is a read-only register.

**Table 10-14. Channel Busy Status Register (CH_BUSY)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 7 | CH_BUSY[7] | RO | Status of the channel 7 BUSY bit of the C7_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |
| 6 | CH_BUSY[6] | RO | Status of the channel 6 BUSY bit of the C6_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |
| 5 | CH_BUSY[5] | RO | Status of the channel 5 BUSY bit of the C5_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |
| 4 | CH_BUSY[4] | RO | Status of the channel 4 BUSY bit of the C4_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |
| 3 | CH_BUSY[3] | RO | Status of the channel 3 BUSY bit of the C3_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |
| 2 | CH_BUSY[2] | RO | Status of the channel 2 BUSY bit of the C2_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |
| 1 | CH_BUSY[1] | RO | Status of the channel 1 BUSY bit of the C1_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |
| 0 | CH_BUSY[0] | RO | Status of the channel 0 BUSY bit of the C0_CFG register<br>0: BUSY = 0<br>1: BUSY = 1 |

Value after reset: 0000h

### 10.4.5.10 Main Configuration Status Register (CSR) (Offset = 0x24)

**Table 10-15. Main Configuration Status Register (CSR)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [7:3] | - | RO | Reserved |
| 2 | M1ENDIAN | RW | AHB Master 1 endian configuration<br>0 = Little-endian<br>1 = Big-endian<br>Reset value = 0 |
| 1 | M0ENDIAN | RW | AHB Master 0 endian configuration<br>0 = Little-endian<br>1 = Big-endian<br>Reset value = 0 |
| 0 | DMACEN | RW | DMA controller enable<br>0 = Disable<br>1 = Enable<br>Reset value = 0 |

Value after reset: 0000h

### 10.4.5.11 Synchronization Register (SYNC) (Offset = 0x28)

The synchronization register is used to disable or enable the DMA controller internal synchronization logic for the hardware DMA request (**dma_req**[15:0]). If the synchronization logic is enabled, it synchronizes hardware DMA request through the synchronization of two clocks. If the synchronization logic is disabled, it synchronizes hardware DMA request bypass synchronization circuit. It is necessary to enable the synchronization logic if the DMA controller and the external device that drives **dma_req** run at different clock domains.

**Table 10-16. Synchronization Register (SYNC)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 7 | SYNC[7] | RW | DMA synchronization logic enable for channel 7 request:<br>0: Disable<br>1: Enable |

| Bit | Name | Access | Description |
|---|---|---|---|
| 6 | SYNC[6] | RW | DMA synchronization logic enable for channel 6 request<br>0: Disable<br>1: Enable |
| 5 | SYNC[5] | RW | DMA synchronization logic enable for channel 5 request<br>0: Disable<br>1: Enable |
| 4 | SYNC[4] | RW | DMA synchronization logic enable for channel 4 request<br>0: Disable<br>1: Enable |
| 3 | SYNC[3] | RW | DMA synchronization logic enable for channel 3 request<br>0: Disable<br>1: Enable |
| 2 | SYNC[2] | RW | DMA synchronization logic enable for channel 2 request<br>0: Disable<br>1: Enable |
| 1 | SYNC[1] | RW | DMA synchronization logic enable for channel 1 request<br>0: Disable<br>1: Enable |
| 0 | SYNC[0] | RW | DMA synchronization logic enable for channel 0 request<br>0: Disable<br>1: Enable |

Value after reset: 0000h

### 10.4.5.12 Channel Control Register (C*n*_CSR)

**Table 10-17. Channel Control Register (C*n*_CSR)**

| Bit | Name | Access | Description |
|---|---|---|---|
| 31 | TC_MSK | RW | Terminal count status mask for current transaction<br>0: When the terminal count happens, the TC status register will be set (Default).<br>1: When the terminal count happens, the TC status register will not be set. |
| [30:27] | - | RO | Reserved |
| [26:24] | DMA_FF_TH | RW | DMA FIFO threshold value<br>000: Threshold value = 1<br>001: Threshold value = 2<br>010: Threshold value = 4<br>011: Threshold value = 8<br>100: Threshold value = 16<br>101 ~ 111: Reserved<br>When the DMA FIFO space ≥ DMA_FF_TH, the DMA controller will start transferring the data from the source to FIFO.<br>When the number of valid data in the DMA FIFO is greater than that of DMA_FF_TH, the DMA controller will start poping out data from FIFO to the destination.<br>Note: DMA_FF_TH should be smaller than 1/2 DMA FIFO size. |
| [23:22] | CHPRI | RW | Channel priority level<br>3: Highest priority<br>2: Second higher priority<br>1: Third higher priority<br>0: Lowest priority (Default) |
| 21 | PROT3 | RW | PROT: Protection information for cacheability[2]<br>0: Not cacheable (Default)<br>1: Cacheable |
| 20 | PROT2 | RW | PROT: Protection information for bufferablity[3]<br>0: Not bufferable (Default)<br>1: Bufferable |

[2] This bit controls the AHB HPROT[3].
[3] This bit controls the AHB HPROT[2].

| Bit | Name | Access | Description |
|---|---|---|---|
| 19 | PROT1 | RW | PROT: Protection information for mode indication[4]<br>0: User mode (Default)<br>1: Privileged mode |
| [18:16] | SRC_SIZE | RW | Source burst size selection<br>000: Burst size = 1 (Default)<br>001: Burst size = 4<br>010: Burst size = 8<br>011: Burst size = 16<br>100: Burst size = 32<br>101: Burst size = 64<br>110: Burst size = 128<br>111: Burst size = 256<br>Notes:<br>The following descriptions only apply to DMAC32:<br>1. The source burst size is not related to the HBURST (AHB signals). It indicates the number of transfers existing before the DMA re-arbitrates among the enabled channels. The number of bytes to be transferred for one burst depends on this source burst size and the source transfer width. For example, if the source burst size is 64 (Bits[18:16] are set as 101) and source transfer width is 16 bits (Bits[13:11] are set as 001), the total number of bytes for this burst transfer will be 128 (64 * 2).<br>2. (Burst size * SRC_WIDTH) must be equal to or larger than DST_WIDTH. Thus, the following settings are not allowed:<br>(For DMAC32 or DMAC64)<br>Burst size = 1, source width = 8, destination width = 16<br>Burst size = 1, source width = 8, destination width = 32<br>Burst size = 1, source width = 16, destination width = 32<br>(For DMAC64)<br>Burst size = 1, source width = 8, destination width = 64<br>Burst size = 1, source width = 16, destination width = 64<br>Burst size = 1, source width = 32, destination width = 64<br>Burst size = 4, source width = 8, destination width = 64. |

[4] This bit controls the AHB HPROT[1].

| Bit | Name | Access | Description |
|---|---|---|---|
| 15 | ABT | WO | Transaction abort<br>Writing '1' to this bit will cause DMA to stop the current transfer. Set the ABT[n] bit of Error/Abort Status Register and assert dmaint interrupt if INT_ABT_MST = 0.<br>Note:<br>No matter what value users write, if users write '1' to bit-15 (ABT), all other bits of this register will remain the same. That is, users cannot program bits[15:1] and the other bits of this register simultaneously. |
| 14 | | RW | Reserved |
| [13:11] | SRC_WIDTH | RW | **DMAC32**<br>Source transfer width<br>The hardware automatically packs and unpacks the data as required.<br>000: Transfer width is 8 bits.<br>001: Transfer width is 16 bits.<br>010: Transfer width is 32 bits (Default).<br>Others: Reserved<br>Note:<br>If the source transfer width < the destination transfer width, DMA will pack the source input data. For example, if the source transfer width is 8 bits and the destination transfer width is 32 bits, DMA will then pack four sets of 8-bit source data and transfer one set of 32-bit data to the destination.<br>Limitation: Do not set SRCAD_CTL = 01 (Decrement source address) when the pack function works; otherwise, DMA will have a wrong action.<br>If the source transfer width > the destination transfer width, DMA will unpack the source input data. For example, if the source transfer width is 32 bits and the destination transfer width is 8 bits, DMA will then unpack the source 32-bit data and transfer four sets of 8-bit data to the destination.<br><br>**DMAC64**<br>Source transfer width<br>The hardware automatically packs and unpacks the data as required.<br>000: Transfer width is 8 bits.<br>001: Transfer width is 16 bits.<br>010: Transfer width is 32 bits.<br>011: Transfer width is 64 bits (Default).<br>Others: Reserved<br>Note:<br>If the source transfer width < the destination transfer width, DMA will pack the source input data. For example, if the source transfer width is 8 bits and the destination transfer width is 32 bits, DMA will then pack four sets of 8-bit source data and transfer one set of 32-bit data to the destination. |

| Bit | Name | Access | Description |
|---|---|---|---|
| | | | Limitation: Do not set SRCAD_CTL = 01 (Decrement source address) when the pack function works; otherwise DMA will have a wrong action.<br><br>If the source transfer width > the destination transfer width, the DMA will unpack the source input data. For example, if the source transfer width is 32 bits and the destination transfer width is 8 bits, DMA will then unpack the source 32-bit data and transfer four sets of 8-bit data to the destination. |
| [10:8] | DST_WIDTH | RW | **DMAC32**<br><br>Destination transfer width<br><br>The hardware automatically packs and unpacks the data as required.<br><br>000: Transfer width is 8 bits.<br><br>001: Transfer width is 16 bits.<br><br>010: Transfer width is 32 bits (Default).<br><br>Others: Reserved<br><br>Note:<br><br>If the source transfer width < the destination transfer width, DMA will pack the source input data. For example, if the source transfer width is 8 bits and the destination transfer width is 32 bits, DMA will then pack four sets of 8-bit source data and transfer one set of 32-bit data to the destination.<br><br>Limitation: Do not set SRCAD_CTL = 01 (Decrement source address) when the pack function works; otherwise DMA will have a wrong action.<br><br>If the source transfer width > the destination transfer width, DMA will unpack the source input data. For example, if the source transfer width is 32 bits and the destination transfer width is 8 bits, DMA will then unpack the source 32-bit data and transfer four sets of 8-bit data to the destination. |

| Bit | Name | Access | Description |
|---|---|---|---|
| | | | **DMAC64**<br>Destination transfer width<br>The hardware automatically packs and unpacks the data as required.<br>000: Transfer width is 8 bits.<br>001: Transfer width is 16 bits.<br>010: Transfer width is 32 bits.<br>011: Transfer width is 64 bits (Default).<br>Others: Reserved<br>Note:<br>If the source transfer width < the destination transfer width, DMA will pack the source input data. For example, if the source transfer width is 8 bits and the destination transfer width is 32 bits, DMA will then pack four sets of 8-bit source data and transfer one set of 32-bit data to the destination.<br>Limitation: Do not set SRCAD_CTL = 01 (Decrement source address) when the pack function works; otherwise DMA will have a wrong action.<br>If the source transfer width > the destination transfer width, DMA will unpack the source input data. For example, if the source transfer width is 32 bits and the destination transfer width is 8 bits, DMA will then unpack the source 32-bit data and transfer four sets of 8-bit data to the destination. |
| 7 | MODE | RW | 0: Normal mode (Default)<br>1: Hardware handshake mode |
| [6:5] | SRCAD_CTL | RW | Source address control<br>00: Increment source address (Default)<br>01: Decrement source address<br>10: Fixed source address<br>11: Reserved<br>Note:<br>If the source transfer width < the destination transfer width, DMA will pack the source input data. For example, if the source transfer width is 8 bits and the destination transfer width is 32 bits, DMA will then pack four sets of 8-bit source data and transfer one set of 32-bit data to the destination.<br>Limitation: Do not set SRCAD_CTL = 01 (Decrement source address) when the pack function works; otherwise, DMA will have a wrong action.<br>If the source transfer width > the destination transfer width, DMA will unpack the source input data. For example, if the source transfer width is 32 bits and the destination transfer width is 8 bits, DMA will then unpack the source 32-bit data and transfer four sets of 8-bit data to the destination. |

| Bit | Name | Access | Description |
|---|---|---|---|
| [4:3] | DSTAD_CTL | RW | Destination Address Control<br>00: Increment destination address (Default)<br>01: Decrement destination address<br>10: Fixed destination address<br>11: Reserved<br>Note:<br>If the source transfer width < the destination transfer width, DMA will pack the source input data. For example, if the source transfer width is 8 bits and the destination transfer width is 32 bits, DMA will then pack four sets of 8-bit source data and transfer one set of 32-bit data to the destination.<br>Limitation: Do not set SRCAD_CTL = 01 (Decrement source address) when the pack function works; otherwise, DMA will have a wrong action.<br>If the source transfer width > the destination transfer width, DMA will unpack the source input data. For example, if the source transfer width is 32 bits and the destination transfer width is 8 bits, DMA will then unpack the source 32-bit data and transfer four sets of 8-bit data to the destination. |
| 2 | SRC_SEL | RW | 0: AHB Master 0 is the source (Default).<br>1: AHB Master 1 is the source. |
| 1 | DST_SEL | RW | 0: AHB Master 0 is the destination (Default).<br>1: AHB Master 1 is the destination. |
| 0 | CH_EN | RW | Channel Enable<br>0: Disable (Default)<br>1: Enable |

Value after reset: 0000_1200h

### 10.4.5.13 Channel Configuration Register (C*n*_CFG)

**Table 10-18. Channel Configuration Register (C*n*_CFG)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:20] | - | RO | Reserved |
| [19:16] | LLP_CNT | RO | Chain transfer counter<br>This counter is reset to 0 when CH_EN changes from 0 to 1.<br>If the chain transfer is enabled, LLP_CNT will be increased by 1 after each block data transfer is finished. |
| [15:14] | - | RO | Reserved |

| Bit | Name | Access | Description |
|---|---|---|---|
| 13 | DST_HE | RW | Destination Hardware Handshake Mode enable<br>0: Disable<br>1: Enable<br>When the destination hardware handshake is disabled, DMA will start transferring data without waiting the destination request.<br>This bit is only valid when DMAC is in the Hardware Handshake mode. |
| [12:9] | DST_RS | RW | Destination DMA request select<br>It specifies which dma_req will be the destination req and will be used only when the DMA Hardware Handshake mode is enabled. |
| 8 | BUSY | RO | The DMA channel is busy.<br>Please note that only one channel is busy at a time. |
| 7 | SRC_HE | RW | Source Hardware Handshake Mode enable:<br>0: Disable<br>1: Enable<br>When the source hardware handshake is disabled, DMA will start transferring data without waiting for the source request.<br>This bit is only valid when DMAC is in the Hardware Handshake mode |
| [6:3] | SRC_RS | RW | Source DMA request select:<br>It specifies the source request, dma_req, which can be used only when the DMA Hardware Handshake mode is enabled. |
| 2 | INT_ABT_MSK | RW | Channel abort interrupt mask<br>0: No mask interrupt<br>1: Mask interrupt (Default) |
| 1 | INT_ERR_MSK | RW | Channel error interrupt mask<br>0: No mask interrupt<br>1: Mask interrupt (Default) |
| 0 | INT_TC_MSK | RW | Channel terminal count interrupt mask<br>0: No mask interrupt<br>1: Mask interrupt (Default) |

Value after reset: 0000_0007h

### 10.4.5.14 Channel Source Address Register (C*n*_SrcAddr)

**Table 10-19. Channel Source Address Register (C*n*_SrcAddr)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:0] | C*n*_SrcAddr | RW | Source starting address<br>Note: As soon as the DMA transaction is completed, its value will change to the DMA source ending address. |

Value after reset: Undefined

### 10.4.5.15 Channel Destination Address Register (C*n*_DstAddr)

**Table 10-20. Channel Destination Address Register (C*n*_DstAddr)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:0] | C*n*_DstAddr | RW | Destination starting address<br>Note: As soon as the DMA transaction is completed, its value will change to the DMA destination ending address. |

Value after reset: Undefined

### 10.4.5.16 Linked List Descriptor Pointer (Cn_LLP)

**Table 10-21. Linked List Descriptor Pointer (Cn_LLP)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:2] | - | RW | Linked list descriptor pointer address |
| 1 | - | RW | Reserved |
| 0 | - | RW | Master for loading the next LLP:<br>0: Load the next LLP from the AHB Master 0 (Default)<br>1: Load the next LLP from the AHB Master 1 |

Value after reset: 0000_0000h

### 10.4.5.17 Transfer Size Register (C*n*_SIZE)

**Table 10-22. Transfer Size Register (C*n*_SIZE)**

| Bit | Name | Access | Description |
|---|---|---|---|
| [31:22] | - | RO | Reserved |
| [21:0] | TOT_SIZE | RW | Total transfer size<br>The transfer unit depends on the source width. For example:<br>SRC_WIDTH = 000, unit: 8-bit<br>SRC_WIDTH = 001, unit: 16-bit<br>SRC_WIDTH = 010, unit: 32-bit<br>SRC_WIDTH = 011, unit: 64-bit<br>Note: When the DMA transaction is done, its value changes to 0. |

Value after reset: Undefined

### 10.4.5.18 DMA Source/Destination Data Transfer Mapping Rule

**Rule 1:** Push into FIFO from the Source Bus Byte Lane according to the following sequence if the source width ≥ the destination width.

Destination width:

Byte: $\{B_n, B_n, B_n, B_n\}$ n = 0 ~ 3 according to the source address (n = address[1:0])

Hword: $\{B_{1+2n}, B_{2n}, B_{1+2n}, B_{2n}\}$ n = 0 ~ 1 according to the source address (n = address[1])

Word: $\{B_3, B_2, B_1, B_0\}$

Please note that the push sequence (n = 0 -> 1 -> 2 -> 3 or 3 -> 2-> 1 -> 0) is dependent on the destination addressing order (Increment or decrement).

**Rule 2:** Push into FIFO from the collected (Packed) Source Bus Byte Lane according to the following sequence if source width < destination width and source address control is the increment type. (The pack function is not allowed as the decrement type.)

If the source width is 8 bits and the destination width is 32 bits, pack the data according to the sequence: $B_{n+0} \rightarrow B_{n+1} \rightarrow B_{n+2} \rightarrow B_{n+3}$ , then push the $\{B_{n+3}, B_{n+2}, B_{n+1}, B_{n+0}\}$ into FIFO.

If the source width is 8 bits and the destination width is 16 bits, pack the data according to the sequence: $B_{n+0}$ → $B_{n+1}$, then push the {$B_{n+1}$, $B_{n+0}$, $B_{n+1}$, $B_{n+0}$}. After that, pack the data according to the sequence: $B_{n+2}$ → $B_{n+3}$, then push the {$B_{n+3}$, $B_{n+2}$, $B_{n+3}$, $B_{n+2}$} into FIFO.

If the source width is 16 bits and the destination width is 32 bits, pack the data according to the sequence: {$B_{n+1}$, $B_{n+0}$} → {$B_{n+3}$, $B_{n+2}$}, then push the {$B_{n+3}$, $B_{n+2}$, $B_{n+1}$, $B_{n+0}$} into FIFO.

**Rule 3:** Pop from FIFO and unpack (If the source width > the destination width) to the Destination Bus Byte Lane according to the following sequence:

Destination Increment:

8: {B0, B0, B0, B0} → {B1, B1, B1, B1} → {B2, B2, B2, B2} → {B3, B3, B3, B3} (Byte Lane)

16: {B1, B0, B1, B0} → {B3, B2, B3, B2} (Byte Lane)

32: {B3, B2, B1, B0} (Byte Lane)

Destination Decrement:

8: {B3, B3, B3, B3} → {B2, B2, B2, B2} → {B1, B1, B1, B1} → {B0, B0, B0, B0} (Byte Lane)

16: {B3, B2, B3, B2} → {B1, B0, B1, B0} (Byte Lane)

32: {B3, B2, B1, B0} (Byte Lane)

Destination Fix (Same as the Increment):

8: {B0, B0, B0, B0} → {B1, B1, B1, B1} → {B2, B2, B2, B2} → {B3, B3, B3, B3} (Byte Lane)

16: {B1, B0, B1, B0} → {B3, B2, B3, B2} (Byte Lane)

32: {B3, B2, B1, B0} (Byte Lane)

## 10.5 Programming Sequence

### 10.5.1 Channel Initialization (Normal Mode, No Chain Transfer)

1. Set CSR
   a. Decide the master 0 interface and the master 1 endianness by setting M1ENDIAN and M0ENDIAN
   b. Enable the DMA controller by setting DMACEN to '1'
2. Set channel registers
   a. Set Cn_CFG to decide which channel's interrupt need to be enabled
   b. Set transfer source address, Cn_SrcAddr, and destination address, Cn_DstAddr
   c. Set Linked List Pointer Cn_LLP to '0'
   d. Set the transfer number, Cn_SIZE, to determine how many transfers are required in a DMA transaction
   e. (If DMA has the bridge configuration option and wants to use this function when DMAC32 is applied) Set Cn_DevRegBase, Cn_DevRegSize, Cn_DevDtBase, and Cn_DevDtSize to let DMA controller assert h1sel_br[n] and h1sel_dma[n] when the access address is located in these ranges
   f. Set Cn_CSR to decide the priority (CHIPRI), transfer burst size (SRC_SIZE), transfer width (SRC_WIDTH, DST_WIDTH), increment or decrement address, and source or destination interface (SRC_SEL, DST_SEL)
   g. Set MODE in the Cn_CSR to the normal mode
3. Start the DMA transfer by setting CH_EN to '1' in the Cn_CSR

### 10.5.2 Channel Initialization (Normal Mode, Chain Transfer)

1. Fill the Link List description according to Table 10-1, Table 10-2, and Table 10-3
2. Set CSR
   a. Decide the master 0 interface and the master 1 endianness by setting M1ENDIAN and M0ENDIAN
   b. Enable the DMA controller by setting DMACEN to '1'
3. Set channel registers (First Link List transfer)
   a. Set Cn_CFG to decide which interrupt of a channel needs to be enabled
   b. Set the transfer source address, Cn_SrcAddr, and the destination address, Cn_DstAddr
   c. Set Link List Pointer, Cn_LLP, to the next starting address of the Link List descriptor
   d. Set transfer number, Cn_SIZE, to determine how many transfers are required in a DMA transaction
   e. (If DMA has the bridge configuration option and wants to use this function when DMAC32 is applied) Set Cn_DevRegBase, Cn_DevRegSize, Cn_DevDtBase, and Cn_DevDtSize to assert h1sel_br[n] and h1sel_dma[n] when the access address is located in these ranges
   f. Set Cn_CSR to decide the priority (CHIPRI), transfer burst size (SRC_SIZE), transfer width (SRC_WIDTH, DST_WIDTH), increment or decrement address, and source or destination interface (SRC_SEL, DST_SEL)
   g. Set MODE in the Cn_CSR to Normal Mode
4. Start the DMA transfer by setting CH_EN to '1' in Cn_CSR

### 10.5.3 Channel Initialization (Hardware Handshake Mode, No Chain Transfer)

1. Set CSR
   a. Decide the master 0 interface and the master 1 endianness by setting M1ENDIAN and M0ENDIAN
   b. Enable the DMA controller by setting DMACEN to '1'
2. Set SYNC to decide whether the synchronization logic of a channel needs to be enabled
3. Set channel registers
   a. Set Cn_CFG to decide which interrupt of a channel needs to be enabled
   b. Set the transfer source address, Cn_SrcAddr, and the destination address, Cn_DstAddr
   c. Set Link List Pointer, Cn_LLP to '0'
   d. Set the transfer number, Cn_SIZE, to determine how many transfers are required in a DMA transaction
   e. (If DMA has the bridge configuration option and wants to use this function when DMAC32 is applied.) Set Cn_DevRegBase, Cn_DevRegSize, Cn_DevDtBase, and Cn_DevDtSize to assert h1sel_br[n] and h1sel_dma[n] when the access address is located in these ranges
   f. Set Cn_CSR to decide the priority (CHIPRI), transfer burst size (SRC_SIZE), transfer width (SRC_WIDTH, DST_WIDTH), increment or decrement address, and sources or the destination interface (SRC_SEL, DST_SEL)
   g. Set MODE in Cn_CSR to the Hardware Handshake Mode
4. Start the DMA transfer by setting CH_EN to '1' in the Cn_CSR

### 10.5.4 Channel Initialization (Hardware Handshake Mode, Chain Transfer)

1. Fill the Link List description according to Table 10-1, Table 10-2, and Table 10-3
2. Set CSR
   a. Decide the master 0 interface and the master 1 endianness by setting M1ENDIAN and M0ENDIAN
   b. Enable the DMA controller by setting DMACEN to '1'
3. Set SYNC to decide whether the synchronization logic of a channel needs to be enabled
4. Set channel registers
   a. Set Cn_CFG to decide which interrupt of a channel needs to be enabled
   b. Set the transfer source address, Cn_SrcAddr, and the destination address, Cn_DstAddr
   c. Set Link List Pointer, Cn_LLP, to the starting address of the Link List descriptor
   d. Set transfer number, Cn_SIZE, to determine how many transfers are required in a DMA transaction
   e. (If DMA has the bridge configuration option and wants to use this function when DMAC32 is applied.) Set Cn_DevRegBase, Cn_DevRegSize, Cn_DevDtBase, and Cn_DevDtSize to assert the h1sel_br[n] and h1sel_dma[n] when the access address is located in these ranges
   f. Set Cn_CSR to decide the priority (CHIPRI), transfer burst size (SRC_SIZE), transfer width (SRC_WIDTH, DST_WIDTH), increment or decrement address, and source or destination interface (SRC_SEL, DST_SEL)
   g. Set MODE in the Cn_CSR to the Hardware Handshake mode
5. Start the DMA transfer by setting CH_EN to 1 in the Cn_CSR

### 10.5.5 Channel Abort (During DMA Data Transfer)

1. Set Cn_CFG to enable to abort interrupt (INT_ABT_MSK)
2. Set (ABT) to abort the current data transfer
3. Assert wait interrupt (dmaint)
4. Read ERR/ABT to confirm which channel has been aborted
5. Set INT_ERR/ABT_CLR to clear ERR/ABT. The channel can be reuse again.

# Chapter 11

# AES-DES Cipher Coprocessor

This chapter contains the following sections:


- 11.1 General Description
- 11.2 Features
- 11.3 Block Diagram
- 11.4 Function Description
- 11.5 Register Descriptions
- 11.6 Programming Sequence

## 11.1 General Description

The AES-DES cipher coprocessor provides an efficient hardware implementation of the DES/Triple-DES/ AES algorithm for high-performance encryption and decryption, which can be applied to a variety of applications.

In DES/Triple-DES configuration, the FTAES020S supports four block cipher modes, including ECB, CBC, CFB, and OFB. In AES configuration, it supports five block cipher modes, including ECB, CBC, CTR, CFB, and OFB.

The host processor interface is compliant with AMBA AHB 2.0. It is available as an IP core for integration into the AMBA system. The AES-DES cipher coprocessor provides DMA function, which can reduce the overhead on the processor for data transfers and, thereby, improve the system performance.

### 11.1.1 Terminology

| Term | Description |
|---|---|
| AES | Advanced Encryption Standard |
| DES | Data Encryption Standard |
| Triple_DES | Triple Data Encryption Standard |
| Block Size | Block size in this document represents the data counts processed during one cipher operation cycle |
| Cipher Operation Cycle | Cycle times between input data and output data of security engine |
| CBC | Cipher Block Chaining Mode |
| ECB | Electronic Codebook Mode |
| CTR | Counter Mode |
| CFB | 8-bit Cipher Feedback Mode |
| OFB | Output Feedback Mode |
| INFIFO | Input FIFO |
| OUTFIFO | Output FIFO |

## 11.2 Features

- AMBA AHB 2.0 Slave/Master compliant, but does not support slave split/retry functions
- DES/Triple-DES encryption/decryption compliant with NIST standard
- AES 128-bit/192-bit/256-bit encryption/decryption compliant with NIST standard
- Supports the block cipher mode
  - DES/Triple-DES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
  - AES
    - ECB mode
    - CBC mode
    - CFB mode
    - OFB mode
    - CTR mode
- Provides DMA function

## 11.3 Block Diagram

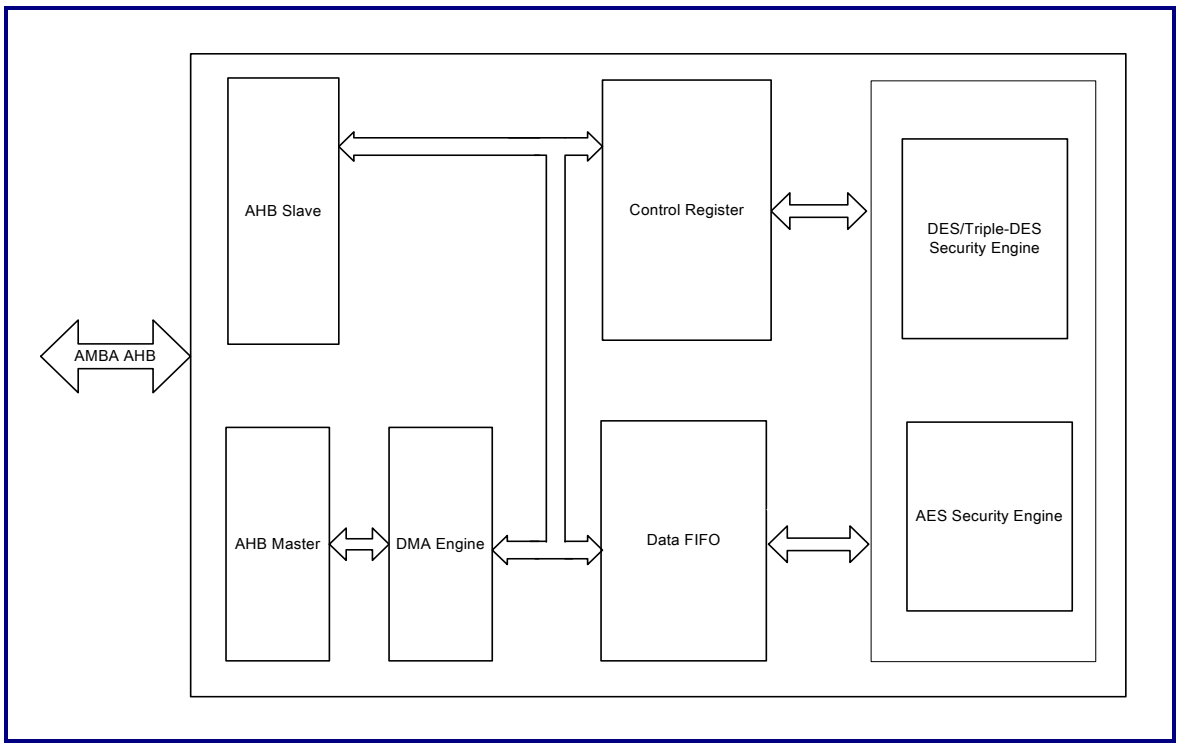



**Figure 11-1. Block Diagram**

## 11.4 Function Description

### 11.4.1 AHB Slave and Control Register

This block provides the AHB bus interface and several sets of status and control registers. Users can program these registers for appropriate application through the AHB bus.

### 11.4.2 AHB Master

The main function of the AHB master is to transfer data through the AHB bus. This design is compliant with the AMBA AHB 2.0 specification. The burst type of the AHB master is an incremental burst of unspecified length (INCR). The data size may be a word or byte, depending on the remaining total transfer size. If the total transfer size is larger than four bytes, the data size will be a word; otherwise, it will be a byte.

### 11.4.3 DMA Engine

The DMA block supports the transferring of data from the source address to the destination address. Users have to program the related registers, such as the source address, destination address, and total transfer size, before enabling DMA. When the DMA engine completes the total transfer size, an interrupt will be triggered. Users can set the DMAStop bit of the DMACtrl register to disable the DMA function. When the DMAStop bit is set, FIFO will be reset, and the DMA stop interrupt will be triggered.

### 11.4.4 Data FIFO

The data FIFO is composed of input FIFO (INFIFO) and output FIFO (OUTFIFO). DMA reads data from the AHB bus and pushes it into INFIFO. DMA pops data from OUTFIFO and writes it to the AHB bus. When the security engine is active, it pops the original data from INFIFO and pushes the result data into OUTFIFO. The status of the data FIFO can be read from the FIFO status register.

### 11.4.5 Security Engine

The security engine supports three cipher algorithms: AES, DES, and Triple-DES, and up to five operation modes for these algorithms: ECB, CBC, CTR, CFB, and OFB. For AES, the engine provides five operation modes (ECB, CBC, CTR, CFB, and OFB) and three key sizes (128 bits, 192 bits, 256 bits), which are determined by register setting. For DES and Triple DES, it provides four operation modes (ECB, CBC, CFB, OFB). The key size of DES is 64 bits. The key size of Triple-DES is 192 bits. Users can program the EncrypControl register to set specific values for different applications. When DMA is enabled and the data in FIFO is ready, the security engine will encrypt or decrypt data from INFIFO and return the resulting data to OUTFIFO. For each encryption/decryption cycle, one block size of data will be processed. It repeats this operation until the total transfer is completed. The block size for each mode is shown below.

**Table 11-1. Block Size of Each Mode**

| Algorithm | AES | DES | AES/DES |
|---|---|---|---|
| Mode | ECB/CBC/OFB/CTR | ECB/CBC/OFB | CFB |
| Block size | 16 bytes | 8 bytes | 1 byte |

Table 11-2 lists the clock cycles of one cipher operation of different algorithms.

**Table 11-2. Cipher Operation Cycle**

| Algorithm | AES | DES | Triple DES |
|---|---|---|---|
| Clock cycle | 12 | 16 | 48 |

## 11.5 Register Descriptions

### 11.5.1 Summary of Control Registers

**Table 11-3. Summary of Control Registers**

| Address (Offset) | Type | Name | Description | Reset Value |
|---|---|---|---|---|
| 0x00 | R/W | EncryptControl | Encryption/decryption control register | 00000000 |
| 0x04 | - | - | Reserved | - |
| 0x08 | R | FIFOStatus | IN/OUT FIFO status information | 00000005 |
| 0x0c | R | PErrStatus | Parity error information | 00FFFFFF |
| 0x10 | R/W | Key0 | The 0$^{th}$ 32 bits of the key stream | 00000000 |
| 0x14 | R/W | Key1 | The 1$^{th}$ 32 bits of the key stream | 00000000 |
| 0x18 | R/W | Key2 | The 2$^{th}$ 32 bits of the key stream | 00000000 |
| 0x1c | R/W | Key3 | The 3$^{th}$ 32 bits of the key stream | 00000000 |
| 0x20 | R/W | Key4 | The 4$^{th}$ 32 bits of the key stream | 00000000 |
| 0x24 | R/W | Key5 | The 5$^{th}$ 32 bits of the key stream | 00000000 |
| 0x28 | R/W | Key6 | The 6$^{th}$ 32 bits of the key stream | 00000000 |
| 0x2c | R/W | Key7 | The 7$^{th}$ 32 bits of the key stream | 00000000 |
| 0x30 | R/W | IV0 | The 0$^{th}$ 32 bits of the initial vector stream | 00000000 |
| 0x34 | R/W | IV1 | The 1$^{st}$ 32 bits of the initial vector stream | 00000000 |
| 0x38 | R/W | IV2 | The 2$^{nd}$ 32 bits of the initial vector stream | 00000000 |
| 0x3c | R/W | IV3 | The 3$^{rd}$ 32 bits of the initial vector stream | 00000000 |
| 0x40 | - | - | Reserved | - |
| 0x44 | - | - | Reserved | - |
| 0x48 | R/W | DMASrc | DMA source address | 00000000 |
| 0x4c | R/W | DMADes | DMA destination address | 00000000 |
| 0x50 | R/W | DMATrasSize | DMA total transfer size | 00000000 |
| 0x54 | R/W | DMACtrl | DMA control register | 00000000 |
| 0x58 | R/W | FIFOThold | FIFO threshold | 00000010 |

| Address (Offset) | Type | Name | Description | Reset Value |
|---|---|---|---|---|
| 0x5c | R/W | IntrEnable | Interrupt enable bit | 00000000 |
| 0x60 | R | IntrSrc | Interrupt source information before mask | 00000000 |
| 0x64 | R | MaskedIntrStatus | Interrupt status after mask | 00000000 |
| 0x64 | W | IntrClr | Interrupt clear register | 00000000 |
| 0x6c | - | - | Reserved | - |
| 0x70 | R | REVSION | Version register | - |
| 0x74 | R | FEATURE | Version register | - |
| 0x78 | - | - | Reserved | - |
| 0x7c | - | - | Reserved | - |
| 0x80 | R | LAST_IV0 | Last block IV output for the next block | - |
| 0x84 | R | LAST_IV1 | Last block IV output for the next block | - |
| 0x88 | R | LAST_IV2 | Last block IV output for the next block | - |
| 0x8c | R | LAST_IV3 | Last block IV output for the next block | - |

### 11.5.2 Register Descriptions

The following sections describe the APB Bridge registers in details.

#### 11.5.2.1 Encryption Control Register (Offset = 0x00 EncryptControl)

This register stores information to control the function of the security engine.

**Table 11-4. Bit Assignments of Encryption Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:9] | - | - | Reserved |
| 8 | pchk | R/W | Parity check bit; only used in the DES mode<br>1: Enable the parity check<br>0: Disable the parity check |
| 7 | first | R/W | This bit specifies the first data block to load the initial vector:<br>When operating in CBC, CTR, CFB, or OFB mode, users should specify the first block to load the initial vector for the cipher operation.<br>1: Specify the first block of cipher operation<br>0: Non-first block<br>This bit automatically returns to 0 after finishing one cipher operation. |

| Bit | Name | Type | Description |
|---|---|---|---|
| [6:4] | emode | R/W | Operation mode select:<br>3'b000: ECB mode<br>3'b001: CBC mode<br>3'b010: CTR mode<br>3'b100: CFB mode<br>3'b101: OFB mode |
| [3:1] | method | R/W | Encryption algorithm select:<br>3'b000: DES<br>3'b001: Triple DES<br>3'b100: AES-128 (Key size = 128 bits)<br>3'b101: AES-192 (Key size = 192 bits)<br>3'b110: AES-256 (Key size = 256 bits) |
| 0 | decrypt | R/W | Decryption or encryption stage:<br>1: Decryption stage<br>0: Encryption stage |

### 11.5.2.2 FIFO Status Register (Offset = 0x08 *FIFOStatus*)

This register stores the FIFO information, which reflects the current entities in INFIFO and OUTFIFO.

**Table 11-5. Bit Assignments of FIFO Status Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | of_entity | R | OUT FIFO entity count |
| [23:16] | If_entity | R | IN FIFO entity count |
| [15:4] | - | - | Reserved |
| 3 | Offull | R | OUT FIFO is full. |
| 2 | ofempty | R | OUT FIFO is empty. |
| 1 | Iffull | R | IN FIFO is full. |
| 0 | ifempty | R | IN FIFO is empty. |

### 11.5.2.3 Parity Error Register (Offset = 0x0c *PErrStatus*)

This register stores the results of the parity check.

**Table 11-6. Bit Assignments of Parity Error Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | perr | R | Parity error |
| [30:24] | - | - | Reserved |
| [23:0] | perrb | R | Parity error bits of each byte of the DES Keys |

### 11.5.2.4 Security Key *N* Registers (Offset = 0x10 ~ 0x2c)

These registers (0x10 ~ 0x2c) store the security key stream information for the cipher operation. For AES-128, the key stream is 128 bits (Key0 ~ Key3). For AES-192, the key stream is 192 bits (Key0 ~ Key5). For AES-256, the key stream is 256 bits (Key0 ~ Key7). For DES, the key stream is 64 bits (Key0 ~Key1). For Triple DES, the key stream is three 64 bits ({Key0, Key1}, {Key2, Key3}, {Key4, Key5}).

**Table 11-7. Bit Assignment of Key *N* Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | Key*N* | R/W | The $N^{th}$ 32 bits of key stream |

### 11.5.2.5 Initial Vector *N* Registers (Offset = 0x30 ~ 0x3c)

These registers (0x30 ~ 0x3c) store the initial vector information for the cipher operation. For AES, the initial vector is 128 bits (IV0 ~ IV3). For DES/Triple-DES, the initial vector is 64 bits (IV0 ~ IV1).

**Table 11-8. Bit Assignment of Initial Vector *N* Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | IV*N* | R/W | The $N^{th}$ 32 bits of Initial Vector stream |

### 11.5.2.6 DMA Source Address Register (Offset = 0x48 DMASrc)

This register stores information about the DMA source address from which DMA reads data into INFIFO for cipher operation.

**Table 11-9. Bit Assignment of DMA Source Address Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | DMASrc | - | Source address of DMA |

### 11.5.2.7 DMA Destination Address Register (Offset = 0x4c DMADes)

This register stores information about the DMA destination address to which DMA writes data from OUTFIFO for the cipher operation.

**Table 11-10. Bit Assignment of DMA Destination Address Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | DMADes | - | Destination address of DMA |

### 11.5.2.8 DMA Transfer Size Register (Offset = 0x50 DMATrasSize)

This register stores information about the total transfer size that the DMA will transfer. The unit of the value is a byte.

**Table 11-11. Bit Assignments of DMA Transfer Size Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:12] | - | - | Reserved |
| [11:0] | TranSize | R/W | Total byte sizes of a DMA transfer in the range of 0 ~ 4095.<br>The byte sizes must be the multiple of the block size, which is determined by the modes of the cipher operation specified in EncryptControl.<br>The block size for all modes of AES, except the CFB mode, is 16 bytes (128 bits).<br>The block size for all modes of DES, except for the CFB mode, is 8 bytes (64 bits).<br>The block size of the CFB mode of AES and DES is 1 byte (8 bits). |

### 11.5.2.9 DMA Control Register (Offset = 0x54 DMACtrl)

The information in this register is used to enable or disable the DMA function. When writing '1' to the DmaStop bit, the encryption/decryption action will be stopped immediately, and the interrupt (StopIntr, Register 0x60[2]) will be asserted. The hardware will automatically set the DmaStop bit to '0' when users clear the interrupt (ClrStopIntr, Register 0x68[2]).

**Table 11-12. Bit Assignments of DMA Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:3] | - | - | Reserved |
| 2 | DmaStop | W | 1: Stop DMA<br>0: No effect |
| | | R | 1: DMA stop operation<br>0: No DMA stop operation |
| 1 | - | - | Reserved |
| 0 | DmaEn | W | 1: Enable DMA<br>0: No effect |
| | | R | 1: DMA enable<br>0: DMA disable |

### 11.5.2.10 FIFO Threshold Register (Offset = 0x58 FIFOThold)

This register stores information about the threshold used as the condition to activate read or write transfers of DMA.

**Table 11-13. Bit Assignments of FIFO Threshold Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | OUTFIFOThold | R/W | Watermark of the OUTFIFO data count that activates the engine to pop data from OUTFIFO and write data into the memory.<br>Unit: Bytes |
| [7:0] | INFIFOThold | R/W | Watermark of the INFIFO data count that activates the engine to read data from the memory and push the data into INFIFO.<br>Unit: Bytes |

### 11.5.2.11 Interrupt Enable Register (Offset = 0x5c IntrEnable)

This register stores information to enable or disable the interrupt source.

**Table 11-14. Bit Assignments of Interrupt Enable Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:3] | - | - | Reserved |
| 2 | StopIntrEn | R/W | 0: Disable the DMA Stop Interrupt source<br>1: Enable the DMA Stop Interrupt source |
| 1 | ErrIntrEn | R/W | 0: Disable the DMA Receives hresp Error Interrupt source<br>1: Enable the DMA Receives hresp Error Interrupt source |
| 0 | DoneIntrEn | R/W | 0: Disable the DMA Transfer Done Interrupt source<br>1: Enable the DMA Transfer Done Interrupt source |

### 11.5.2.12 Interrupt Source Register (Offset = 0x60 IntrSrc)

This register stores information about the interrupt source.

**Table 11-15. Bit Assignments of Interrupt Status Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:3] | - | - | Reserved |
| 2 | StopIntr | R | DMA Stop Interrupt source before mask |
| 1 | ErrIntr | R | DMA Receives hresp Error Interrupt source before mask |
| 0 | DoneIntr | R | DMA Transfer Done Interrupt source before mask |

### 11.5.2.13 Masked Interrupt Status (Offset = 0x64 *MaskedIntrStatus*)

This register stores information about the interrupt after mask.

**Table 11-16. Bit Assignments of Masked Interrupt Status Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:3] | - | - | Reserved |
| 2 | StopIntrMsk | R | DMA Stop Interrupt after mask |
| 1 | ErrIntrMsk | R | DMA Receive hresp Error after mask |
| 0 | DoneIntrMsk | R | DMA Transfer Done Interrupt after mask |

### 11.5.2.14 Interrupt Clear Register (Offset = 0x68 IntrClr)

This register is used to clear the interrupt.

**Table 11-17. Bit Assignments of Interrupt Clear Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:3] | - | - | Reserved |
| 2 | ClrStopIntr | w | 0: No effect<br>1: Clear the DMA Stop Interrupt |
| 1 | ClrErrIntr | w | 0: No effect<br>1: Clear the DMA Receive hresp Error Interrupt |
| 0 | ClrDoneIntr | w | 0: No effect<br>1: Clear the DMA Transfer Done Interrupt |

### 11.5.2.15 Revision Register (Offset = 0x70 REVISION)

This register stores the revision information.

**Table 11-18. Bit Assignment of Revision Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:3] | REVISION | R | The revision number |

### 11.5.2.16 Feature Register (Offset = 0x74 FEATURE)

This register stores information of hardware feature.

**Table 11-19. Bit Assignment of Feature Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | FIFO_DEPTH | R | The IN/OUT FIFO depth. Unit: Byte |

### 11.5.2.17 Last Initial Vector *N* Register (Offset = 0x80 ~ 0x8c)

These registers (0x80 ~ 0x8c) store information about the initial vectors of the last block size for the cipher operation. For AES, the initial vector is 128 bits (IV0 ~ IV3). For DES/Triple-DES, the initial vector is 64 bits (IV0 ~ IV1). For some modes, such as the EBC mode or IV2 ~ IV3 for DES/Triple-DES, the register value is '0'.

**Table 11-20. Bit Assignment of Last Initial Vector *N* Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | Last_IV *N* | R | The $N^{th}$ 32 bits of the Last Initial Vector stream |

## 11.6 Programming Sequence

The programming sequence is as follows:

1. Set the EncryptControl register for the encryption/decryption operations
2. Set Initial Vector IV to a mode of a cipher algorithm, such as CBC, CTR, CFB, or OFB. There are four 32-bit registers (IV0 ~ IV3) for the initial vector setting. For the AES algorithm, IV is 16 bytes (IV0 ~ IV3). For the DES algorithm, IV is 8 bytes (IV0 ~ IV1). One IVn register is four bytes of the initial vector. The byte sequence is in the most-significant-bit-first order. The initial vectors for different cipher algorithms are shown below:
   DES/Triple-DES: IV[63:0] = {IV0, IV1}
   AES: IV[127:0] = {IV0, IV1, IV2, IV3}

**Table 11-21. Byte Sequence of Initial Vector**

| IV Register | IV0 | | | | IV1 | | | | IV2 | | | | IV3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte sequence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | … |
| IV[63:0] | Initial vector | | | | | | | | | | | | | |

3. Set the key values for different cipher algorithms, such as AES, DES, and Triple DES. There are eight 32-bit registers (Key0 ~ Key7) for the key value setting. For the AES-128 algorithm, the key value is 16 bytes (Key0 ~ Key3). For the AES-192 algorithm, the key value is 24 bytes (Key0 ~ Key5). For the AES256 algorithm, the key value is 32 bytes (Key0 ~ Key7). For the DES algorithm, the key value is bytes (Key0 ~ Key1). For the triple DES algorithm, there are three 8-byte ({Key0, Key1}, {Key2, Key3}, {Key4, Key5}) key values. The byte sequence is in the most-significant-bit-first order. The lists below are the key values for different cipher algorithms:

**AES-128 Algorithm**

AES-128Key[127:0] = {Key0, Key1, Key2, Key3}

AES-128Key[127:96] = Key0

AES-128Key[95:64] = Key1

AES-128Key[63:0] = Key2

AES-128Key[31:0] = Key3

**Table 11-22. AES-128 Key Stream of Byte Sequence**

| Key Register | IV0 | | | | IV1 | | | | IV2 | | | | IV3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte Sequence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | … |
| Key[127:0] | AES-128 key value | | | | | | | | | | | | | |

**AES-192 Algorithm**

AES-192Key[191:0] = {Key0, Key1, Key2, Key3, Key4, Key5}

AES-192Key[191:160] = Key0

AES-192Key[159:128] = Key1

AES-192Key[127:96 ] = Key2

AES-192Key[95:64] = Key3

AES-192Key[63:0] = Key4

AES-192Key[31:0] = Key5

**Table 11-23. AES-192 Key Stream of Byte Sequence**

| Key Register | Key0 | | | | Key1 | | | | Key2 | | | | | Key5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte Sequence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | … | … | 23 |
| Key[191:0] | AES-192 key value | | | | | | | | | | | | | |

### AES-256 Algorithm

AES-256Key[255:0] = {Key0, Key1, Key2, Key3, Key4, Key5, Key6, Key7}

AES-256Key[255:224] = Key0

AES-256Key[223:192] = Key1

AES-256Key[191:160] = Key2

AES-256Key[159:128] = Key3

AES-256Key[127:96 ] = Key4

AES-256Key[95:64] = Key5

AES-256Key[63:0] = Key6

AES-256Key[31:0] = Key7

**Table 11-24. AES-256 Key Stream of Byte Sequence**

| Key Register | Key0 | | | | Key1 | | | | Key2 | | | | | Key7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte Sequence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | … | … | 31 |
| Key[255:0] | AES-256 key value | | | | | | | | | | | | | |

### DES Algorithm

DESkey[63:0] = {Key0, Key1}

DESkey[63:32] = Key0

DESkey[31:0] = Key1

**Table 11-25. DES Key Stream of Byte Sequence**

| Key Register | Key0 | | | | Key1 | | | | Key2 | | | | Key3 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte Sequence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | … |
| Key[63:0] | DES key value | | | | | | | | Reserved | | | | | |

### Triple DES Algorithm

Triple-DESKey1[63:0] = {Key0, Key1}

Triple-DESKey1[63:32] = Key0

Triple-DESkey1[31:0] = Key1

Triple-DESKey2[63:0] = {Key2, Key3}

Triple-DESKey2[63:32] = Key2

Triple-DESKey2[31:0] = Key3

Triple-DESKey3[63:0] = {Key4, Key5}

Triple-DESKey3[63:32] = Key4

Triple-DESKey3[31:0] = Key5

**Table 11-26. Triple-DES Key Stream of Byte Sequence**

| Key Register | Key0 | | | | Key1 | | | | Key2 | | | | Key5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte Sequence | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | … | … | 23 |
| Key[191:0] | Triple-DES Key1 | | | | | | | | Triple-DES Key2 | | | | Triple-DES Key3 | |

1. Locate the memory space of the input and output data blocks. Prepare the input data block in a specific memory space. The cipher or plain text is represented in the byte sequence order, such as b0, b1, b2, b3, and b4. The index indicates an address offset. For example, b0 indicates data b0 resides in offset 0 and b1 resides in offset 1. The endianness in this design is little endian. In order for the system to function correctly, all modules accessing the memory should be of the same endianness.
2. Set DMA related registers

   Set the DMASrc register for source address, DMADes for destination address, and DMATrasSize for total data transfer size from source to destination. The value of DMATrasSize represents the byte count and should be a multiple of the block size that is determined by the cipher algorithm.
3. Set the FIFOThold register to specify the FIFO threshold for DMA to activate the read/write transfers from the memory. Users can set two watermarks in the FIFOThold register as the condition to activate the read or write transfers. The read transfer meets the condition that the entities of INFIFO are less than the value of INFIFOThold or one block size. The write transfer meets the condition that the entities of OUTFIFO are more than the value of OUTFIFOThold or one block size.
4. Set the IntrEnable register to control the interrupt signal
5. Set the DmaEn bit of DMACtrl to '1' to activate the DMA engine and the security engine
6. After the total transfer size is complete, interrupt will be triggered and DMA will be stopped.

# Chapter 12

# USB 2.0 OTG Controller

This chapter contains the following sections:


- 12.1 General Description
- 12.2 Features
- 12.3 Block Diagram
- 12.4 Register Definition

## 12.1 General Description

The Universal Serial Bus (USB) 2.0 On-The-Go (OTG) controller plays dual roles that act as a host controller or act as a peripheral controller. When acting as a host controller, it contains a USB host controller that supports transactions at all speeds. Without the software intervention, the host controller can deal with a transaction-based data structure to offload the CPU and automatically transmit and receive data on the USB bus. When acting as a peripheral controller, each endpoint, except for the endpoint 0, accepts the programmable HS/FS transfer types to suit all applications. In addition, complying with the OTG standards means that both Session Request Protocol (SRP) and Host Negotiation Protocol (HNP) are supported. The system bus can be a PVCI or AHB 32-bit bus interface. The transceiver interface is UTMI+ Level 3, which supports the high-speed, full-speed, and low-speed transfers with a high-speed or full-speed hub.

## 12.2 Features

- Compliant with USB Specification, revision 2.0
- Compliant with On-The-Go supplement to USB 2.0 Specification, revision 1.0
- Supports UTMI+ Level 3 compliant transceiver
- Compatible with EHCI 1.0
- Built-in PVCI-compatible system bus and optional AMBA AHB bus interface
- Supports OTG SRP and HNP protocols
- Supports point-to-point communications with one high-speed, full-speed, or low-speed device
- Supports high-speed or full-speed hub
- Hardware configurable endpoints as HS/FS device
- Supports isochronous/interrupt/control/bulk transfers in host and device modes
- Compatible with EHCI data structures (FSTN and SITD back points are not supported)
- Supports embedded DMA access to FIFO
- Supports SRAM interface for FIFO
- Supports suspend, remote wake-up, and resume functions
- Compatible with Faraday FZOTG230 transceiver
- Completely synchronous design
- USB OTG software subsystem

## 12.3 Block Diagram

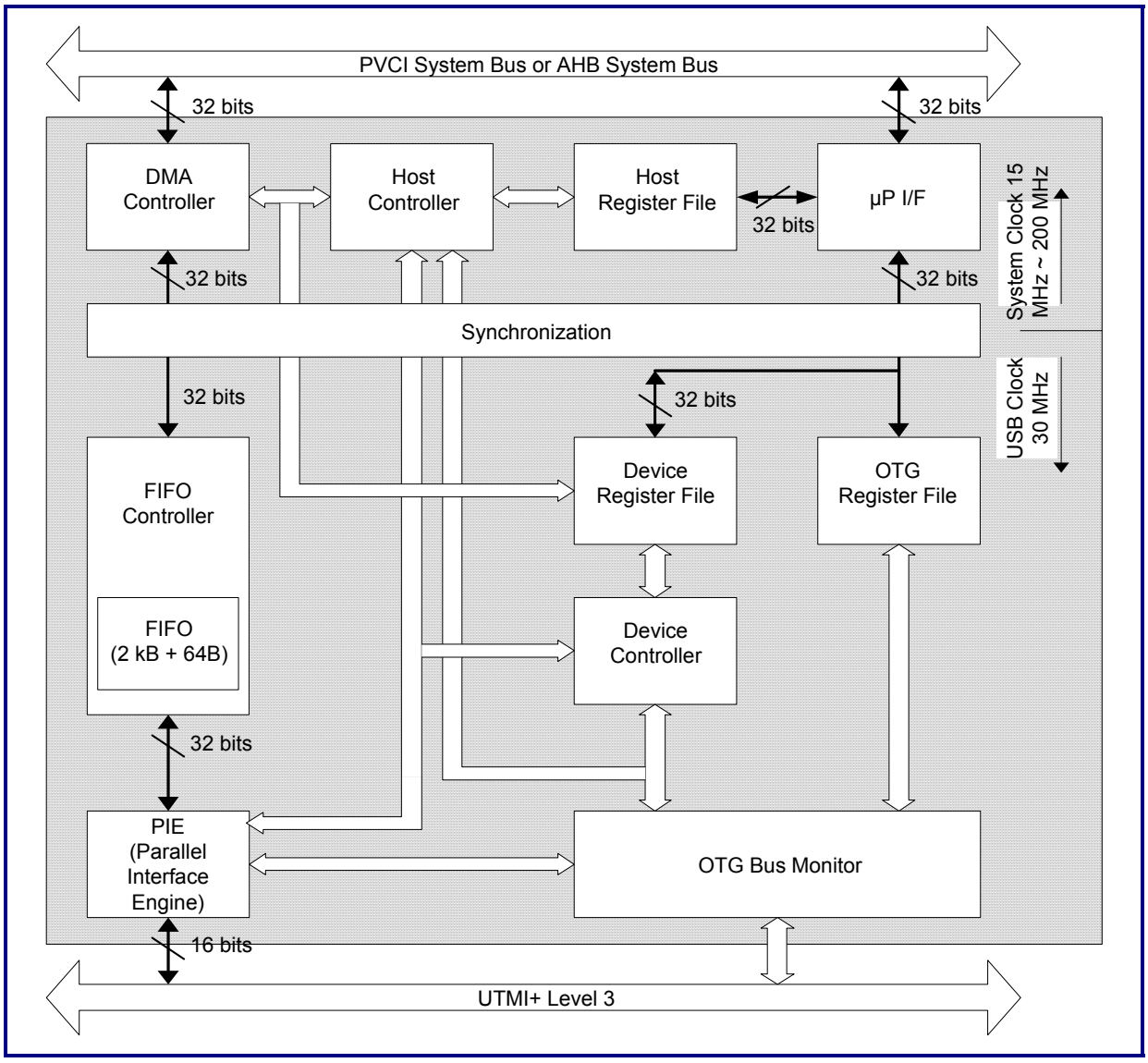



**Figure 12-1. Block Diagram**

## 12.4 Register Definition

The register definitions provide four major parts of the OTG 2.0 controller. The first part is the registers of the host controller. The second part is the registers of the OTG controller, which can be set to the OTG command state machine. The third part is the global registers to the configure settings of the whole controller. The last part is the registers of a device. The detailed information on the control registers is described in the following subsections.

The following notations are used to describe the register access types:

| | |
|---|---|
| RO | Read Only<br>If the register is read only, writing on this register will have no effect. |
| R/W | Read/Write<br>A register with this attribute can be read and written. Please note that individual bit in some read/write register may be read only. |
| R/WC | Read/Write Clear<br>A register bit with this attribute will be read and written. However, writing '1' will clear (Set to '0') the corresponding bit, whereas writing '0' has no effect. |

For a device controller, the following table shows the default values of the reset conditions:

| | |
|---|---|
| Default Reset | Power-on reset or hardware reset<br>Each register is reset to its default value. |
| (U) | Bus reset<br>USB bus reset. Not all registers use the bus reset. |
| (S) | Soft reset<br>By setting bit 4 of the Main Control register (0x100) of the device, software will reset the SOF Frame Number register (0x10C) and the SOF Mask Timer register (0x110). |

### 12.4.1 Register Summary

**Table12-1. Register Summary**

| Name | Address Range | Size (Byte) | Description |
|---|---|---|---|
| HC registers | 9'h000 ~ 9'h07F | 128 | Host controller register space |
| OTG registers | 9'h080 ~ 9'h0BF | 64 | OTG controller register space |
| Global registers | 9'h0C0 ~ 9'h0FF | 64 | Global register space |
| Device registers | 9'h100 ~ 9'h1FF | 256 | Device controller register space |

### 12.4.1.1 Host Controller Registers (Address = 0x000 ~ 0x07F)

#### 12.4.1.1.1 HC Capability Register (Address = 0x000)

**Table12-2. HC Capability Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:16] | HCIVERSION | RO | 16'h0100 | **Host Controller Interface Version Number**<br>This field contains a BCD encoding of the EHCI revision number supported by the host controller. |
| [15:8] | Reserved | RO | 8'h0 | - |
| [7:0] | CAPLENGTH | RO | 8'h10 | **Capability Register Length**<br>This field is used as an offset that can add to the register base to find out the beginning of Operational Register Space. |

#### 12.4.1.1.2 HCSPARAMS – HC Structural Parameters (Address = 0x004)

**Table12-3. HC Structural Parameters**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:4] | Reserved | RO | 28'b0 | - |
| [3:0] | N_PORTS | RO | 4'h1 | **Number of Ports**<br>This field specifies the number of the physical downstream ports implemented on the host controller. |

#### 12.4.1.1.3 HCCPARAMS – HC Capability Parameters (Address = 0x008)

**Table 12-4. HC Capability Parameters**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:3] | Reserved | RO | 29'b0 | - |
| 2 | ASYN_SCH_PARK_CAP | RO | 1'b1 | **Asynchronous Schedule Park Capability**<br>The host controller supports the park feature for the high-speed queue heads in the asynchronous schedule. This feature can be disabled or enabled and set to a specific level by using the Asynchronous Schedule Park Mode Enable and Asynchronous Schedule Park Mode Count fields in the USBCMD register. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 1 | PROG_FR_LIST_FLAG | RO | 1'b1 | **Programmable Frame List Flag**<br>When this bit is set to '1b', the system software specifies and uses a smaller frame list to configure the host controller via the Frame List Size field of the USBCMD register. This requirement ensures that the frame list is always physically contiguous. |
| 0 | Reserved | RO | 1'b0 | - |

### 12.4.1.1.4 USBCMD – HC USB Command Register (Address = 0x010)

**Table 12-5. HC USB Command Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:24] | Reserved | RO | 8'b0 | - |
| [23:16] | INT_THRC | R/W | 8'h08 | **Interrupt Threshold Control**<br>This field is used by the system software to select the maximum rate at which the host controller will issue interrupt. The only valid values are defined as below:<br>Value — Maximum interrupt interval for high speed<br>00h — Reserved<br>01h — 1 micro frame<br>02h — 2 micro frames<br>04h — 4 micro frames<br>08h — 8 micro frames (Default, equals to 1 ms)<br>10h — 16 micro frames (2 ms)<br>20h — 32 micro frames (4 ms)<br>40h — 64 micro frames (8 ms)<br>Note: During the full-speed mode, these registers are all reserved. |
| [15:12] | Reserved | RO | 4'b0 | - |
| 11 | ASYN_PK_EN | R/W | 1'b1 | **Asynchronous Schedule Park Mode Enable**<br>The software uses this bit to enable or disable the Park mode. When this bit is set to '1', the park mode is enabled. |
| 10 | Reserved | RO | 1'b0 | - |
| [9:8] | ASYN_PK_CNT | R/W | 2'b11 | **Asynchronous Schedule Park Mode Count**<br>This field contains the count of number of successive transactions the host controller is allowed to execute from a high-speed queue head on the asynchronous schedule. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | Reserved | RO | 1'b0 | - |
| 6 | INT_OAAD | R/W | 1'b0 | **Interrupt on Asynchronous Advance Doorbell**<br>This bit is used as a doorbell by the software to inform the host controller to issue an interrupt the next time it advances asynchronous schedule. |
| 5 | ASCH_EN | R/W | 1'b0 | **Asynchronous Schedule Enable**<br>This bit controls the host controller to skip processing the Asynchronous Schedule.<br>0: Do not process the asynchronous schedule.<br>1: Use ASYNCLISTADDR register to access the Asynchronous Schedule. |
| 4 | PSCH_EN | R/W | 1'b0 | **Periodic Schedule Enable**<br>This bit controls the host controller to skip processing the Periodic Schedule.<br>0: Do not process Periodic Schedule.<br>1: Use PERIODICKISTBASE register to access the Periodic Schedule. |
| [3:2] | FRL_SIZE | R/W | 2'b00 | **Frame List Size**<br>This field specifies the size of the frame list.<br>00: 1024 elements (4096 bytes, the default value)<br>01: 512 elements (2048 bytes)<br>10: 256 elements (1024 bytes)<br>11: Reserved |
| 1 | HC_RESET | R/W | 1'b0 | **Host Controller Reset**<br>This control bit is used by software to reset the host controller. |
| 0 | RS | R/W | 1'b0 | **Run/Stop**<br>When this bit is set to '1b', the host controller proceeds with the execution of schedule.<br>0: Stop<br>1: Run |

### 12.4.1.1.5 USBSTS – HC USB Status Register (Address = 0x014)

**Table 12-6. HC USB Status Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:16] | Reserved | RO | 16'b0 | - |
| 15 | ASCH_STS | RO | 1'b0 | **Asynchronous Schedule Status**<br>This bit reports the actual status of Asynchronous Schedule. |
| 14 | PSCH_STS | RO | 1'b0 | **Periodic Schedule Status**<br>This bit reports the actual status of Periodic Schedule. |
| 13 | Reclamation | RO | 1'b0 | **Reclamation**<br>This is a read-only status bit, which is used to detect an empty Asynchronous Schedule. |
| 12 | HCHalted | RO | 1'b1 | **Host Controller Halted**<br>This bit is a zero whenever the Run/Stop bit is set to '1'. The host controller will set this bit to 1b after it has stopped executing, as a result of the Run/Stop bit being set to 0b. |
| [11:6] | Reserved | RO | 6'b0 | - |
| 5 | INT_OAA | R/WC | 1'b0 | **Interrupt on Async. Advance**<br>This status bit indicates the assertion of interrupt on Async. Advance Doorbell. |
| 4 | H_SYSERR | R/WC | 1'b0 | **Host System Error**<br>The host controller sets this bit to '1' when a serious error occurred during a host system access involving host controller module. |
| 3 | FRL_ROL | R/WC | 1'b0 | **Frame List Rollover**<br>The host controller sets this bit to '1' when Frame List Index changes from its maximum value to '0'. |
| 2 | PO_CHG_DET | R/WC | 1'b0 | **Port Change Detect**<br>The host controller sets this bit to'1' when any port has a change bit transition from '0' to '1'. This bit is loaded with OR of all the PORTSC change bits. |
| 1 | USBERR_INT | R/WC | 1'b0 | **USB Error Interrupt**<br>The host controller sets this bit to '1' when the completion of a USB transaction results in an error condition. |
| 0 | USB_INT | R/WC | 1'b0 | **USB Interrupt**<br>The host controller sets this bit to '1' upon the completion of a USB transaction. |

### 12.4.1.1.6 USBINTR – HC USB Interrupt Enable Register (Address = 0x018)

**Table 12-7. HC USB Interrupt Enable Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:6] | Reserved | RO | 26'b0 | - |
| 5 | INT_OAA_EN | R/W | 1'b0 | **Interrupt on Async. Advance Enable**<br>When this bit is '1' and the Interrupt on Async. Advance bit in the USBSTS register is also '1', the host controller will issue an interrupt at the next interrupt threshold. |
| 4 | H_SYSERR_EN | R/W | 1'b0 | **Host System Error Enable**<br>When this bit is '1' and the Host System Error Status bit in the USBSTS register is also '1', the host controller will issue an interrupt. |
| 3 | FRL_ROL_EN | R/W | 1'b0 | **Frame List Rollover Enable**<br>When this bit is '1' and the Frame List Rollover bit in the USBSTS register is also '1', the host controller will issue an interrupt. |
| 2 | PO_CHG_INT_EN | R/W | 1'b0 | **Port Change Interrupt Enable**<br>When this bit is '1' and the Port Change Detect bit in the USBSTS register is also '1', the host controller will issue an interrupt. |
| 1 | USBERR_INT_EN | R/W | 1'b0 | **USB Error Interrupt Enable**<br>When this bit is a 1 and the USBERRINT bit in USBSTS register is a 1, the host controller will issue an interrupt at the next interrupt threshold. |
| 0 | USB_INT_EN | R/W | 1'b0 | **USB Interrupt Enable**<br>When this bit is '1' and the USBINT bit in the USBSTS register is also '1', the host controller will issue an interrupt at the next interrupt threshold. |

### 12.4.1.1.7 FRINDEX – HC Frame Index Register (Address = 0x01C)

**Table 12-8. HC Frame Index Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:14] | Reserved | RO | 18'b0 | - |
| [13:0] | FRINDEX | R/W | 14'b0 | **Frame Index**<br>This register is used by the host controller to index the frame into the periodic frame list. It updates every 125 microseconds. This register cannot be written unless the host controller is in the Halted state. |

#### 12.4.1.1.8 PERIODICLISTBASE – HC Periodic Frame List Base Address Register (Address = 0x024)

**Table 12-9. HC Periodic Frame List Base Address Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:12] | PERI_BASADR | R/W | Undefined | **Periodic Frame List Base Address**<br>This 32-bit register contains the beginning address of the periodic frame list in the system memory. These bits correspond to memory address signals[31:12]. |
| [11:0] | Reserved | RO | 12'b0 | - |

#### 12.4.1.1.9 ASYNCLISTADDR – HC Current Asynchronous List Address Register (Address = 0x028)

**Table 12-10. HC Current Asynchronous List Address Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:5] | Async_ladr | R/W | Undefined | **Current Asynchronous List Address**<br>This 32-bit register contains the address of the next asynchronous queue head to be executed. These bits correspond to memory address signals[31:5]. |
| [4:0] | Reserved | RO | 5'b0 | - |

#### 12.4.1.1.10 PORTSC – HC Port Status and Control Register (Address = 0x030)

**Table 12-11. HC Port Status and Control Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:21] | Reserved | RO | 11'b0 | - |
| 20 | HC_TST_PKDONE | R/W | 1'b0 | **Data Transfer is Done for Test Packet in Host Port Test Control**<br>The firmware has completely sent the whole test patterns to FIFO for a PHY test packet mode by writing a '1' to this bit. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [19:16] | PORT_TEST | R/W | 4'b0000 | **Port Test Control**<br>When this field is zero, the port is NOT operating in the test mode. A non-zero value indicates that it is operating in the test mode and the specific test mode is indicated by a specific value. The encoding of the test mode bits are (0110b-1111b are reserved):<br>**Bits** **Test Mode**<br>0000b Test mode not enabled<br>0001b Test J_STATE<br>0010b Test K-STATE<br>0011b Test SE0_NAK<br>0100b Test Packet<br>0101b Test FORCE_ENABLE<br>Please note when set to 0100b (Test packet), the test packet must be filled into FIFO by DMA first, and then set HC_TST_PKDONE to 1. |
| [15:12] | Reserved | RO | 4'b0 | - |
| [11:10] | LINE_STS | RO | 2'bxx | **Line State**<br>These bits reflect the current logical levels of the D+ and D- signal lines.<br>**Bits[11:0]** **USB State**<br>00b SE0<br>10b J-State<br>01b K-State<br>11b Undefined |
| 9 | Reserved | RO | 1'b0 | - |
| 8 | PO_RESET | R/W | 1'b0 | **Port Reset**<br>1 = The port is reset.<br>0 = The port is not reset.<br>When the software writes a 1 to this bit, the bus reset sequence defined in the USB spec. is started. The software writes a 0 to this bit to terminate the bus reset sequence. The software must keep this bit at one long enough to ensure the reset sequence.<br>Note: Before setting this bit, the RUN/STOP bit should be set to 0. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | PO_SUSP | R/W | 1'b0 | **Port Suspend**<br>1 = The port is at the suspend state.<br>0 = The port is not at the suspend state.<br>The Port Enable bit and the Suspend bit of this register define the port states as follows:<br>**Bits [Port Enable, Suspend]** — **Port State**<br>0X — Disable<br>10 — Enable<br>11 — Suspend<br>At the suspend state, the downstream propagation of data is blocked on this port, except the port reset. At the suspend state, the port is sensitive to the resume detection.<br>The host controller ignores a zero written to this bit. The host controller will unconditionally set this bit to a zero when:<br>• The software sets the Force Port Resume bit to '0' (From a '1').<br>• The software sets the Port Reset bit to '1' (From a '0').<br>Note: Before setting this bit, the RUN/STOP bit should be set to '0'. |
| 6 | F_PO_RESM | R/W | 1'b0 | **Force Port Resume**<br>1 = Resume detected/driven on port<br>0 = No resume detected/driven on port<br>Software sets this bit to a one to resume signaling. The host controller sets this bit to a one if a J-to-K transition is detected while the port is in the suspend state. When this bit transits to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to a one. |
| [5:4] | Reserved | RO | 2'b0 | - |
| 3 | PO_EN_CHG | R/WC | 1'b0 | **Port Enable/Disable Change**<br>0 = No change<br>1 = Port enabled/disabled status is changed. |
| 2 | PO_EN | R/W | 1'b0 | **Port Enable/Disable**<br>0 = Disable<br>1 = Enable<br>Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 1 | CONN_CHG | R/WC | 1'b0 | **Connect Status Change**<br>0 = Do not change the current status<br>1 = Change the current connect status<br>It indicates that the current connect status is changed on the port. |
| 0 | CONN_STS | RO | 1'b0 | **Current Connect Status**<br>0 = No device is present.<br>1 = Device is present on the port.<br>This bit reflects the current state of the port, and may not correspond directly to the event that setting the connect status change bit. When TST_FORCEEN is set to '1', this signal is the output of u_hdiscon. |

### 12.4.1.1.11 Miscellaneous Register (Address = 0x040)

**Table 12-12. HC Misc. Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:11] | Reserved | RO | 25'b0 | - |
| [10:7] | R_HPROT | R/W | 4'b0011 | User Definition of AHB HPROT<br>Reserved for user control of the AHB HPROT signal.<br>This field exists only when "FOTG210_IF_AHB" is enabled using hardware configuration. Otherwise, this register is reserved as 4'b0 and read only. |
| 6 | HostPhy_Suspend | R/W | 1'b0 | **Host Transceiver Suspend Mode**<br>Active high<br>Place the transceiver in the suspend mode that draws minimal power from power supplies<br>This bit is only used in the host mode and is in the system clock domain instead of the UCLK clock domain. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [5:4] | EOF2_Time | R/W | 2'b0 | **EOF 2 Timing Points**<br>These bits control EOF2 timing point before the next SOF.<br>**High-speed EOF2 time**<br>00b 2 clocks (30 MHz) = 66 ns<br>01b 4 clocks (30 MHz) = 133 ns<br>10b 8 clocks (30 MHz) = 266 ns<br>11b 16 clocks (30 MHz) = 533 ns<br>**Full-speed EOF2 time**<br>00b 20 clocks (30 MHz) = 666 ns<br>01b 40 clocks (30 MHz) = 1.33 µs<br>10b 80 clocks (30 MHz) = 2.66 µs<br>11b 160 clocks (30 MHz) = 5.33 µs<br>**Low-speed EOF2 time**<br>00b 40 clocks (30 MHz) = 1.33 µs<br>01b 80 clocks (30 MHz) = 2.66 µs<br>10b 160 clocks (30 MHz) = 5.33 µs<br>11b 320 clocks (30 MHz) = 10.66 µs |
| [3:2] | EOF1_Time | R/W | 2'b0 | **EOF 1 Timing Points**<br>These bits control EOF1 timing point before the next SOF. They should be adjusted according to the maximum packet size.<br>**High-speed EOF1 time**<br>00b 540 clocks (30 MHz) = 18 µs<br>01b 360 clocks (30 MHz) = 12 µs<br>10b 180 clocks (30 MHz) = 6 µs<br>11b 720 clocks (30 MHz) = 24 µs<br>**Full-speed EOF1 time**<br>00b 1600 clocks (30 MHz) = 53.3 µs<br>01b 1400 clocks (30 MHz) = 46.6 µs<br>10b 1200 clocks (30 MHz) = 40 µs<br>11b 21000 clocks (30 MHz) = 700 µs<br>**Low-speed EOF1 time**<br>00b 3750 clocks (30 MHz) = 125 µs<br>01b 3500 clocks (30 MHz) = 116 µs<br>10b 3250 clocks (30 MHz) = 108 µs<br>11b 4000 clocks (30 MHz) = 133 µs |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [1:0] | ASYN_SCH_SLPT | R/W | 2'b01 | **Asynchronous Schedule Sleep Timer**<br>This field controls the sleep timer of the asynchronous schedule.<br>00b 5 µs<br>01b 10 µs<br>10b 15 µs<br>11b 20 µs |

### 12.4.1.2 OTG Controller Register (Address = 0x080 ~ 0x0BF)

#### 12.4.1.2.1 OTG Control Status Register (Address = 0x080)

**Table 12-13. OTG Control Status Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:24] | - | RO | 8'b0 | Reserved and read as zeros |
| [23:22] | HOST_SPD_TYP | RO | 2'b00 | **Host Speed Type**<br>These bits indicate the speed type when the OTG device functions as the host.<br>2'b10: HS<br>2'b00: FS<br>2'b01: LS<br>2'b11: Reserved |
| 21 | ID | RO | 1 | **Current ID**<br>This bit records the current ID of FOTG210.<br>0: A-device<br>1: B-device |
| 20 | CROLE | RO | 1 | **Current Role**<br>This bit records the current role of FOTG210.<br>0: Host<br>1: Device |
| 19 | VBUS_VLD | RO | 0 | **A-device Vbus Valid**<br>This bit indicates whether or not the voltage on $V_{bus}$ is above the A-device $V_{bus}$ valid threshold. |
| 18 | A_SESS_VLD | RO | 0 | **A-device Session Valid**<br>This bit indicates whether or not the voltage on $V_{bus}$ is above the A-device session valid threshold. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 17 | B_SESS_VLD | RO | 0 | **B-Device Session Valid**<br>This bit indicates whether or not the voltage on Vbus is above the B-device session valid threshold. |
| 16 | B_SESS_END | RO | 1 | **B-Device Session End**<br>This bit indicates whether or not the voltage on Vbus is below the B-device session end threshold. |
| [15:12] | - | R/W | 4'b0 | Reserved for internal testing only. These bits should remain '0.' |
| 11 | HDISCON_FLT_SEL | R/W | 0 | This bit selects a timer to filter out the HDISCON noise from UTMI+.<br>0: Approximated to 135 µs<br>1: Approximated to 270 µs |
| 10 | VBUS_FLT_SEL | R/W | 0 | This bit selects a timer to filter out VBUS_VLD noise from UTMI+.<br>0: Approximated to 135 µs<br>1: Approximated to 472 µs |
| 9 | ID_FLT_SEL | R/W | 1'b0 | This bit selects a timer to filter out ID noise from UTMI+.<br>0: Approximated to 3 ms<br>1: Approximated to 4 ms |
| 8 | A_SRP_RESP_TYP | R/W | 0 | **SRP Response Type**<br>This bit determines the SRP type to which the A-device should respond.<br>0: A-device responds to $V_{bus}$ pulsing.<br>1: A-device responds to data-line pulsing. |
| 7 | A_SRP_DET_EN | R/W | 0 | **A-device SRP Detection Enable**<br>This bit controls whether or not the A-device should detect SRP from the B-device.<br>0: SRP detection is disabled.<br>1: SRP detection is enabled. |
| 6 | A_SET_B_HNP_EN | R/W | 0 | This bit indicates that the current role is A-device and the HNP function of B-device had been enabled. This bit should be set and cleared by software.<br>0: No effect<br>1: HNP feature of B-device had been enabled.<br>This bit is valid while the current role is A-device. This bit will be cleared to a '0' only after the A-device issues a USB reset. |
| 5 | A_BUS_DROP | R/W | 1 | **A-device Bus Drop**<br>This bit determines if the A-device wants to power-down $V_{bus}$. Writing this bit to a 1 will clear BUS_REQ of A-device. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4 | A_BUS_REQ | R/W | 0 | **A-device Bus Request**<br>This bit determines if A-device should take control of the bus.<br>0: A-device stops driving $V_{bus}$ and bus traffic.<br>1: A-device drives $V_{bus}$ and generates bus traffic. |
| 3 | - | RO | 0 | Reserved, and read as zeros. |
| 2 | B_DSCHRG_VBUS | R/W | 0 | B-device discharges Vbus<br>This bit is used to determine if discharging $V_{bus}$ is required after Vbus pulsing during SRP.<br>0: $V_{bus}$ will not be discharged after $V_{bus}$ pulsing.<br>1: $V_{bus}$ will be discharged for 50 ms after $V_{bus}$ pulsing. |
| 1 | B_HNP_EN | R/W | 0 | This bit indicates the B-device has been enabled to perform HNP. This bit can be cleared only after the B-device is reset by host.<br>0: Disable HNP<br>1: Enable HNP |
| 0 | B_BUS_REQ | R/W | 0 | **B-device Bus Request**<br>This bit determines if the B-device should take control of the bus. After SRP pulsing is finished, this bit is automatically cleared by hardware.<br>0: B-device stops driving $V_{bus}$ and generating bus traffic.<br>1: B-device requests to take control of bus. |

### 12.4.1.2.2 OTG Interrupt Status Register (Address = 0x084)

This register defines the interrupt status of FOTG210. The interrupt status is not masked by interrupt enable. That is, if the corresponding events happen, the corresponding status bit is set to '1' even if the corresponding interrupt enable bit is '0.'

**Table 12-14. OTG Interrupt Status Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:13] | - | RO | 0 | Reserved and read as zeros |
| 12 | APLGRMV | R/WC | 0 | **Mini-A Plug Remove**<br>This bit is set to '1' once the mini-A plug is removed. Writing '1' clears this bit while writing '0' has no effect. |
| 11 | BPLGRMV | R/WC | 0 | **Mini-B Plug Remove**<br>This bit is set to'1' once the mini-B plug is removed. Writing '1' clears this bit while writing '0' has no effect. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 10 | OVC | R/WC | 0 | **Over Current Detection**<br>This bit is set to '1' when $V_{bus}$ does not reach VBUS_VLD within the expected time. Writing '1' clears this bit while writing '0' has no effect. This bit is valid only when the current role is A-device. |
| 9 | IDCHG | R/WC | 0 | **ID Change**<br>This bit is set to '1' when the current ID of FOTG210 changes either from A-device to B-device or from B-device to A-device. Writing '1' clears this bit while writing '0' has no effect. |
| 8 | RLCHG | R/WC | 0 | **Role Change**<br>This bit is set to '1' when the current role of FOTG210 changes from host to peripheral or from peripheral to host. Writing '1' clears this bit while writing '0' has no effect. |
| 7 | - | RO | 0 | Reserved, and read as zeros |
| 6 | B_SESS_END | R/WC | 1 | **B_Sess_End**<br>This register is set to '1' when B_SESS_END is high. Writing '1' clears this register while writing '0' has no effect. |
| 5 | A_VBUS_ERR | R/WC | 0 | **A-device Vbus Error**<br>This bit is set to '1' when the OTG state machine moves to "VBUS_ERROR" state. Writing '1' clears this bit while writing '0' has no effect. |
| 4 | A_SRP_DET | R/WC | 0 | **A-device Detects SRP from B-Device**<br>This bit is set to '1' when the A-device detects SRP from B-device. Writing '1' clears this bit while writing '0' has no effect. |
| [3:1] | - | RO | 0 | Reserved, and read as zeros |
| 0 | B_SRP_DN | R/WC | 0 | **B-Device SRP Done**<br>This bit is set to '1' after the B-device has completed the SRP signaling. Writing '1' clears this bit while writing '0' has no effect. |

### 12.4.1.2.3 OTG Interrupt Enable Register (Address = 0x088)

This register defines the interrupt enable of interrupt events. This register will not mask the interrupt status. It only masks the interrupt generation.

**Table 12-15. OTG Interrupt Enable Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:13] | - | RO | 0 | Reserved, and read as zeros |
| 12 | APLGRMV_EN | R/W | 0 | APLGRMV interrupt enable |
| 11 | BPLGRMV_EN | R/W | 0 | BPLGRMV interrupt enable |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 10 | OVC_EN | R/W | 0 | OVC interrupt enable |
| 9 | IDCHG_EN | R/W | 0 | IDCHG interrupt enable |
| 8 | RLCHG_EN | R/W | 0 | RLCHG interrupt enable |
| [7:6] | - | RO | 0 | Reserved, and read as zeros |
| 5 | A_VBUS_ERR_EN | R/W | 0 | A_VBUS_ERR interrupt enable |
| 4 | A_SRP_DET_EN | R/W | 0 | A_SRP_DET interrupt enable |
| [3:1] | - | RO | 0 | Reserved, and read as zeros |
| 0 | B_SRP_DN_EN | R/W | 0 | B_SRP_DN interrupt enable |

### 12.4.1.3 Global Controller Registers (Address = 0x0C0 ~ 0x0FF)

#### 12.4.1.3.1 HC/OTG/DEV Interrupt Status Register (Address = 0x0C0)

**Table 12-16. Global HC/OTG/DEV Interrupt Status Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:3] | - | R | 29'b0 | Reserved, and read as zeros |
| 2 | HC_INT | R/WC | 1'b0 | **HC Interrupt**<br>This bit is set to '1' to indicate that an interrupt is issued from the host controller block. |
| 1 | OTG_INT | R/WC | 1'b0 | **OTG Interrupt**<br>This bit is set to '1' to indicate that an interrupt is issued from the OTG controller block. |
| 0 | DEV_INT | R/WC | 1'b0 | **Device Interrupt**<br>This bit is set to '1' to indicate that an interrupt is issued from the device controller block. |

#### 12.4.1.3.2 Mask of HC/OTG/DEV Interrupt (Address = 0x0C4)

**Table 12-17. Global Mask of HC/OTG/DEV Interrupt**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:4] | Reserved | - | - | |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3 | INT_POLARITY | R/W | 1'b0 | This bit controls the polarity of the system interrupt signal, sys_int_n. The default is active low.<br>0: Active low (Default)<br>1: Active high |
| 2 | MHC_INT | R/W | 1'b0 | This bit masks the interrupt bits of the HC interrupt.<br>0: Enable corresponding interrupt<br>1: Disable corresponding interrupt |
| 1 | MOTG_INT | R/W | 1'b0 | This bit masks the interrupt bits of the OTG interrupt.<br>0: Enable corresponding interrupt<br>1: Disable corresponding interrupt |
| 0 | MDEV_INT | R/W | 1'b0 | This bit masks the interrupt bits of the device interrupt.<br>0: Enable corresponding interrupt<br>1: Disable corresponding interrupt |

### 12.4.1.4 Device Controller Registers (Address = 0x100 ~ 0x1FF)

#### 12.4.1.4.1 Device Main Control Register (Address = 0x100)

**Table 12-18. Device Main Control Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:10] | Reserved | - | - | - |
| 9 | FORCE_FS | R/W | 1'b0 | **Force the device to full-speed**<br>Setting this bit to 1'b1 will disable the high-speed negotiation when reset from USB host starts, thus forces FOTG200 to full–speed in device mode. If this bit is 1'b0, the FOTG200 remains high-speed in the device mode. Please note that the user can only modify this bit before the unplug bit is set; otherwise, it will cause unexpected results. |
| [7:8] | Reserved | - | - | - |
| 6 | HS_EN | RO | 1'b0 | **High-Speed Status**<br>1: The device is in the high-speed mode.<br>0: The device is in the full-speed mode. |
| 5 | CHIP_EN | R/W | 1'b1 | **Chip Enable**<br>Writing a '1' enables the write cycle of FIFO controller (Do not write '0' while in normal operation mode). |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4 | SFRST | R/W | 1'b0 | **Device Software Reset**<br>Writing '1' sets the software-initiated reset of the FOTG210 device. This bit cannot be set when FOTG210 is in the suspend mode, because u_clk is stopped. Setting this bit will cause the de-assertion of the pw_save output if it is asserted. The Chirp sequence will be terminated when this bit is set. The command FIFO will be cleared by setting this bit. The Frame Number register and the SOF Timer Mask register will also be cleared. Please note that the data FIFO status will not be cleared (Software reset is self-cleared). |
| 3 | GOSUSP | R/W | 1'b0 | **Go Suspend**<br>Writing '1' activates the suspend mode of PHY. |
| 2 | GLINT_EN | R/W | 1'b0 | **Global Interrupt Enable**<br>Writing '1' enables all interrupts. Individual interrupt is masked by setting the corresponding bits in the interrupt mask register (Index 0x11 ~ 0x17). |
| 1 | HALF_SPEED | R/W | 1'b0 | **Half Speed Enable**<br>1: The FIFO controller asserts ACK to DMA for every two clock cycles.<br>0: The FIFO controller asserts ACK to DMA continuously.<br>Set this bit to '1' to perform the FPGA implementation |
| 0 | CAP_RMWAKUP | R/W | 1'b0 | **Capability of Remote Wakeup**<br>Writing '1' indicates that FOTG210 has the capability of waking up by using the "wakeup" signal. |

### 12.4.1.4.2 Device Address Register (Address = 0x104)

**Table 12-19. Device Address Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:8] | Reserved | - | - | - |
| 7 | AFT_CONF | R/W | 1'b0<br>1'b0(U) | **After Set Configuration**<br>Writing '1' indicates that the device has successfully executed the SET_CONFIGURATION command. The FOTG210 device will not response to any non-control transfer before this bit is set. |
| [6:0] | DEVADR | R/W | 7'b0<br>7'b0(U) | **Device Address:**<br>This field records the latest USB device address for each SET_ADDRESS. |

### 12.4.1.4.3 Device Test Register (Address = 0x108)

**Table 12-20. Device Test Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:7] | Reserved | - | - | - |
| 6 | DISGENSOF | R/W | 1'b0 | **Disable the generation of SOF**<br>Always use SOF issued by host instead of self-generating SOF |
| 5 | TST_MOD | R/W | 1'b0 | **Test Mode**<br>Writing '1' turns on this bit. When this bit is set to '1', FOTG210 will enter the test mode.<br>In the normal mode, FOTG210 uses a large counter of 10 ms to detect a USB reset.<br>In the test mode, FOTG210 will use a smaller counter to detect the USB reset to save the test cycles on the test machine. |
| 4 | TST_DISTOG | R/W | 1'b0 | **Disable Toggle Sequence**<br>Setting this bit to '1' disables the toggle sequence |
| 3 | TST_DISCRC | R/W | 1'b0 | **Disable CRC**<br>By setting this bit to '1', FOTG210 will not append CRC for the upstream packets. |
| 2 | TST_CLREA | R/WC | 1'b0 | **Clear External Side Address**<br>Writing '1' then '0' clears the external side address for the loop-back test (This bit is self-clear). |
| 1 | TST_LPCX | R/W | 1'b0 | **Loop-back Test for Control Endpoint**<br>Writing '1' indicates the loop-back test for the control transfers. |
| 0 | TST_CLRFF | R/W | 1'b0 | **Clear FIFO**<br>Writing '1' clears all FIFO counters and location counters of PAM (This bit is self-clear). |

### 12.4.1.5 Device SOF Frame Number Register (Address = 0x10C)

**Table 12-21. Device SOF Frame Number Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:14] | Reserved | - | - | - |
| [13:11] | USOFN | RO | 3'b0<br>3'b0(S) | **SOF Micro Frame Number Bits[2:0]**<br>This field records the micro frame number of the high-speed mode. |
| [10:0] | SOFN | RO | 11'b0<br>11'b0(S) | **SOF Frame Number Bits[10:0]**<br>This field records the frame number of the high-speed and full-speed modes. |

### 12.4.1.5.1 Device SOF Mask Timer Register (Address = 0x110)

**Table 12-22. Device SOF Mask Timer Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:16] | Reserved | - | - | - |
| [15:0] | SOFTM | R/W | 16'h44C | **SOF Mask Timer** |
| | | | 16'h44C(S) | Time count since the last SOF in the 30-MHz clock bit. |

### 12.4.1.5.2 PHY Test Mode Selector Register (Address = 0x114)

**Table 12-23. PHY Test Mode Selector Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:5] | Reserved | - | - | - |
| 4 | TST_PKT | R/W | 1'b0 | **Test Mode for Packet**<br>Upon writing a '1' to this bit, FOTG210 repetitively sends the packet defined in the UTMI specification to the transceiver. After the set_feature command shows the test mode and index Test_Packet have been decoded, this bit will be asserted. |
| 3 | TST_SE0NAK | R/W | 1'b0 | Upon writing a '1,' the D+/D- lines are set to HS, quiescent state. The device only responds to a valid HS IN token and always responds to the IN token with NAK. |
| 2 | TST_KSTA | R/W | 1'b0 | Upon writing a '1,' the D+/D- lines are set to the high-speed K state. |
| 1 | TST_JSTA | R/W | 1'b0 | Upon writing a '1,' the D+/D- lines are set to the high-speed J state. |
| 0 | UNPLUG | R/W | 1'b1<br>1'b0(U) | When UNPLUG is set to '1', the device controller will set PHY in the Non-Driving mode to emulate the detachment of a device even when it is already plugged. The USB host will not detect plugging of a device. Such an event is called soft-detachment.<br>After a hardware reset, the UNPLUG will be '1' and therefore the device is soft-detached. When the USB host detects the attachment of a device, the PHY must drive D+ and D- in the manner defined in USB specification. In order to let PHY drive D+ and D-, AP should clear UNPLUG after the hardware reset. If the AP does not clear UNPLUG bit, the device will be always soft-detached and the USB host will never detect the attachment of the device. |

### 12.4.1.5.3 Device Vendor-specific I/O Control Register (Address = 0x118)

**Table 12-24. Device Vendor-specific I/O Control Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:6] | Reserved | - | - | - |
| 5 | VCTLOAD_N | R/W | 1'b1 | **Vendor-Specific Test Mode Control Load**<br>This bit controls the active-low output, u_vctload_n, to PHY. Writing '1' to this bit will set the u_vctload_n output to '1'. When this bit is cleared, the u_vctload_n output will be set to '0'. |
| [4:0] | VCTL | R/W | 5'b0 | **Vendor-Specific Test Mode Control**<br>The programmed value is delivered to PHY via the "u_vctl" output. |

### 12.4.1.5.4 Device CX Configuration Status Register (Address = 0x11C)

**Table 12-25. Device CX Configuration Status Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:8] | Reserved | - | - | - |
| [7:0] | VSTA | RO | Depending on the reset value of PHY | **Vendor-Specific Test Mode Status** |

### 12.4.1.5.5 Device CX Configuration and FIFO Empty Status Register (Address = 0x120)

**Table 12-26. Device CX Configuration and FIFO Empty Status Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | Reserved | - | - | - |
| [30:24] | CX_FNT | RO | 7'b0 | **CX FIFO Byte Count** |
| [23:12] | Reserved | - | - | - |
| [11:8] | F_EMP | RO | 4'b1111 | **FIFO EMPTY**<br>This field set FW to check if FIFO is fully empty. |
| [7:6] | Reserved | - | - | - |
| 5 | CX_EMP | RO | 1'b1 | **CX FIFO is Empty**<br>Setting '1' to this bit indicates that the endpoint 0 FIFO is empty. |
| 4 | CX_FUL | RO | 1'b0 | **CX FIFO is Full**<br>Setting '1' to this bit indicates that the endpoint 0 FIFO is full. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3 | CX_CLR | R/W | 1'b0 | **Clear CX FIFO Data**<br>Writing '1' clears data in the endpoint 0 FIFO. For endpoint 0, all data in FIFO will be cleared no matter whether the previous SETUP or IN or OUT transaction has been completed or not (This bit is self-clear). |
| 2 | CX_STL | R/W | 1'b0<br>1'b0(U) | **Stall CX**<br>Writing '1' to this bit stalls endpoint 0. The stall status will be cleared with the next setup transaction. This bit will be cleared automatically when the transaction of endpoint 0 is finished. Upon detecting the bus reset, the firmware should clear this bit. (When setting this bit, CX_DONE must be set in the same write operation) |
| 1 | TST_PKDONE | R/W | 1'b0 | **Data Transfer is Done for Test Packet**<br>The firmware has completely sent the whole test patterns to endpoint 0 FIFO for a PHY test by writing '1' to this bit. This bit is cleared by a hardware reset. |
| 0 | CX_DONE | R/W | 1'b0<br>1'b0(U) | **Data Transfer is Done for CX**<br>The firmware has finished the whole packet transaction for endpoint 0 by writing '1' to this bit. This bit is cleared by a hardware reset or by using the p_endcx or p_comfail internal signal. |

### 12.4.1.5.6 Device Idle Counter Register (Address = 0x124)

**Table 12-27. Device Idle Counter Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:3] | Reserved | - | - | - |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [2:0] | IDLE_CNT | R/W | 3'b0 | This field controls the timing delay from the time indicated in the GOSUSP bit of the main control register to the time the device entered a suspend mode. The timing delay is denoted as $t_{susp\_delay}$ below.<br>000: $t_{susp\_delay}$ = 0 ms<br>001: $t_{susp\_delay}$ = 1 ms<br>010: $t_{susp\_delay}$ = 2 ms<br>011: $t_{susp\_delay}$ = 3 ms<br>100: $t_{susp\_delay}$ = 4 ms<br>101: $t_{susp\_delay}$ = 5 ms<br>110: $t_{susp\_delay}$ = 6 ms<br>111: $t_{susp\_delay}$ = 7 ms<br>Note: The USB 2.0 specifications define $T_{SUSP}$ so that the device will enter the suspend mode no later than 10 ms after D+/D- is continuously at the idle state. The firmware programmer should be cautious in programming the value of $t_{susp\_delay}$. |

### 12.4.1.5.7 Device Mask of Interrupt Group Register (Address = 0x130)

**Table 12-28. Device Mask of Interrupt Group Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:3] | Reserved | - | - | - |
| 2 | MINT_G2 | R/W | 1'b0 | Mask interrupt of source group 2<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 1 | MINT_G1 | R/W | 1'b0 | Mask interrupt of source group 1<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 0 | MINT_G0 | R/W | 1'b0 | Mask interrupt of source group 0<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |

### 12.4.1.5.8 Device Mask of Interrupt Source Group 0 Register (Address = 0x134)

**Table 12-29. Device Mask of Interrupt Source Group 0 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:6] | Reserved | - | - | - |
| 5 | MCX_COMABORT_INT | R/W | 1'b0 | Mask interrupt of control transfer command abort<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 4 | MCX_COMFAIL_INT | R/W | 1'b0 | Mask interrupt of host that emits extra IN or OUT data interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 3 | MCX_COMEND | R/W | 1'b0 | Mask the host that ends a command (Entering the status stage) interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 2 | MCX_OUT_INT | R/W | 1'b0 | Mask the interrupt bits of endpoint 0 for OUT<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 1 | MCX_IN_INT | R/W | 1'b0 | Mask the interrupt bits of endpoint 0 for IN<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 0 | MCX_SETUP_INT | R/W | 1'b0 | Mask endpoint 0 setup data received interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |

### 12.4.1.5.9 Device Mask of Interrupt Source Group 1 Register (Address = 0x138)

**Table 12-30. Device Mask of Interrupt Source Group 1 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:20] | Reserved | - | - | - |
| 19 | MF3_IN_INT | R/W | 1'b1 | Mask the IN interrupt bits of FIFO 3<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 18 | MF2_IN_INT | R/W | 1'b1 | Mask the IN interrupt bits of FIFO 2<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 17 | MF1_IN_INT | R/W | 1'b1 | Mask the IN interrupt bits of FIFO 1<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 16 | MF0_IN_INT | R/W | 1'b1 | Mask the IN interrupt bits of FIFO 0<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| [15:8] | Reserved | - | - | - |
| 7 | MF3_SPK_INT | R/W | 1'b1 | Mask the Short Packet interrupt of FIFO 3<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 6 | MF3_OUT_INT | R/W | 1'b1 | Mask the OUT interrupt of FIFO 3<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 5 | MF2_SPK_INT | R/W | 1'b1 | Mask the Short Packet interrupt of FIFO 2<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 4 | MF2_OUT_INT | R/W | 1'b1 | Mask the OUT interrupt of FIFO 2<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 3 | MF1_SPK_INT | R/W | 1'b1 | Mask the Short Packet interrupt of FIFO 1<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 2 | MF1_OUT_INT | R/W | 1'b1 | Mask the OUT interrupt of FIFO 1<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 1 | MF0_SPK_INT | R/W | 1'b1 | Mask the Short Packet interrupt of FIFO 0<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 0 | MF0_OUT_INT | R/W | 1'b1 | Mask the OUT interrupt of FIFO 0<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |

### 12.4.1.5.10 Device Mask of Interrupt Source Group 2 Register (Address = 0x13C)

**Table 12-31. Device Mask of Interrupt Source Group 2 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:11] | Reserved | - | - | - |
| 10 | MDev_Wakeup_byVBUS | R/W | 1'b0 | **Mask Dev_Wakeup_byVBUS interrupt**<br>Mask the active Dev_Wakeup_byVBUS interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 11 | MDev_Idle | R/W | 1'b0 | **Mask Dev_Idle interrupt**<br>Mask the active Dev_Idle interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 8 | MDMA_ERROR | R/W | 1'b0 | **Mask DMA Error Interrupt**<br>Mask the active DMA Error interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 7 | MDMA_CMPLT | R/W | 1'b0 | **Mask DMA Completion Interrupt**<br>Mask the active DMA Completion interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 6 | MRX0BYTE_INT | R/W | 1'b0 | **Mask Received Zero-length Data Packet Interrupt**<br>Mask the active Received Zero-length Data Packet interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 5 | MTX0BYTE_INT | R/W | 1'b0 | **Mask Transferred Zero-length Data Packet Interrupt**<br>Mask the active to Transferred Zero-length Data Packet interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 4 | MSEQ_ABORT_INT | R/W | 1'b0 | **Mask ISO Sequential Abort Interrupt**<br>Mask the active ISO Sequential Abort interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3 | MSEQ_ERR_INT | R/W | 1'b0 | **Mask ISO Sequential Error Interrupt**<br>Mask the active Received ISO Sequential Error interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 2 | MRESM_INT | R/W | 1'b0 | **Mask Resume Interrupt**<br>Mask the active Resume State Change Interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 1 | MSUSP_INT | R/W | 1'b0 | **Mask Suspend Interrupt**<br>Mask the active Suspend State Change Interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |
| 0 | MUSBRST_INT | R/W | 1'b0 | **Mask USB Reset Interrupt**<br>Mask the Bus Reset Interrupt bit<br>0: Enable the corresponding interrupt<br>1: Disable the corresponding interrupt |

### 12.4.1.5.11 Device Interrupt Group Register (Address = 0x140)

**Table 12-32. Device Interrupt Group Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:3] | Reserved | - | - | - |
| 2 | INT_G2 | RO | 1'b0 | This bit indicates interrupts occurred in Group 2. |
| 1 | INT_G1 | RO | 1'b0 | This bit indicates interrupts occurred in Group 1. |
| 0 | INT_G0 | RO | 1'b0 | This bit indicates interrupts occurred in Group 0. |

### 12.4.1.5.12 Device Interrupt Source Group 0 Register (Address = 0x144)

**Table 12-33. Device Interrupt Source Group 0 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:6] | Reserved | - | - | - |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 5 | CX_COMABT_INT | R/W | 1'b0 | This bit indicates that a command abort event has happened. For interrupts recorded in this source register, the command abort interrupt has the highest priority. For a command abort interrupt, AP should only clear the CX_COMABT_INT bit. All other operations are unnecessary and should be avoided. In general, the command abort interrupt will be accompanied by CX_SETUP_INT. AP should serve the command abort interrupt to clear CX_COMABT_INT, because CXF FIFO is frozen for AP to access when CX_COMABT_INT remains as '1'. To get 8-byte for SETUP of the command abort, AP should clear CX_COMABT_INT first. |
| 4 | CX_COMFAIL_INT | RO | 1'b0 | This bit indicates that the control transfer has been abnormally terminated.<br><br>This bit will be asserted when the device receives extra IN/OUT token at data stage of control transfer.<br><br>Once this bit is asserted, it will stay at '1' before AP sets the CX_STL bit of the CX_Config_Status register. After setting the CX_STL bit of the CX_Config_Status register, AP should set the CX_DONE bit of the CX_Config_Status register. |
| 3 | CX_COMEND_INT | RO | 1'b0 | This bit indicates that the control transfer has entered the status stage.<br><br>This bit will remain asserted before the firmware sets the CX_DONE bit of the CX Configuration and Status register (Address 0E0, bit 0). |
| 2 | CX_OUT_INT | RO | 1'b0 | This bit indicates that the control transfer contains valid data for control-write transfers.<br><br>This bit will remain asserted until the firmware starts to read data from control transfer FIFO (CXF) of device. |
| 1 | CX_IN_INT | RO | 1'b0 | This bit indicates that the firmware should write data of the control-read transfer to control the transfer FIFO. For control reads with a length of less than or equal to 64 bytes, this bit will never be asserted. The firmware will decode the 8-byte data sent at the SETUP stage of the control transfer. The firmware should write first payload of data into control transfer FIFO if the 8-byte indicates control-read transfer without the assertion of this bit. This bit will be asserted only when the length of control-read transfer is greater than 64 bytes and the USB host has successfully received the data of previous packet. For example, for a 65-byte control-read transfer, the firmware should automatically write the first 64 bytes after it decodes 8 byte of the SETUP data. The firmware will be interrupted to write the 65$^{th}$ byte when the USB host ACKs to the first 64 bytes.<br><br>This bit will remain asserted until FW starts to write data into control transfer FIFO of device. |
| 0 | CX_SETUP_INT | RO | 1'b0 | This bit will remain asserted until the firmware starts to read data from control transfer FIFO of device. |

### 12.4.1.5.13 Device Interrupt Source Group 1 Register (Address = 0x148)

**Table 12-34. Device Interrupt Source Group 1 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:20] | Reserved | - | - | - |
| 19 | F3_IN_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 3 is ready to be written.<br>This bit is cleared under the following two conditions:<br>• A maximum-size packet is received in FIFO 3.<br>• End of DMA transaction |
| 18 | F2_IN_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 2 is ready to be written.<br>This bit is cleared under two conditions:<br>• A maximum-size packet is received in FIFO 2.<br>• End of DMA transaction |
| 17 | F1_IN_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 1 is ready to be written.<br>This bit is cleared under two conditions:<br>• A maximum-size packet is received in FIFO 1.<br>• End of DMA transaction |
| 16 | F0_IN_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 0 is ready to be written.<br>This bit is cleared under two conditions:<br>• A maximum-size packet is received in FIFO 0.<br>• End of DMA transaction |
| [15:8] | Reserved | - | - | - |
| 7 | F3_SPK_INT | RO | 1'b0 | This bit becomes '1' to indicate that the short packet data are received in FIFO 3.<br>This bit is cleared once DMA master reads FIFO 3. |
| 6 | F3_OUT_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 3 is ready to be read.<br>This bit is cleared when all data in FIFO 3 are read out. |
| 5 | F2_SPK_INT | RO | 1'b0 | This bit becomes '1' to indicate that the short packet data are received in FIFO 2.<br>This bit is cleared once DMA master reads FIFO 2. |
| 4 | F2_OUT_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 2 is ready to be read.<br>This bit is cleared when all data in FIFO 2 are read out. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3 | F1_SPK_INT | RO | 1'b0 | This bit becomes '1' to indicate that the short packet data are received in FIFO 1.<br>This bit is cleared once DMA master reads FIFO 1. |
| 2 | F1_OUT_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 1 is ready to be read.<br>This bit is cleared when all data in FIFO 1 are read out. |
| 1 | F0_SPK_INT | RO | 1'b0 | This bit becomes '1' to indicate that the short packet data are received in FIFO 0.<br>This bit is cleared once DMA master reads FIFO 0. |
| 0 | F0_OUT_INT | RO | 1'b0 | This bit becomes '1' to indicate that FIFO 0 is ready to be read.<br>This bit is cleared when all data in FIFO 0 are read out. |

### 12.4.1.5.14 Device Interrupt Source Group 2 Register (Address = 0x14C)

**Table 12-35. Device Interrupt Source Group 2 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:11] | Reserved | - | - | - |
| 10 | Dev_Wakeup_byVBUS | RO | 1'b0 | **Device wakeup by $V_{bus}$ Interrupt**<br>The device is at the idle state and the $V_{bus}$ signal is high, this bit is set to 1'b1. |
| 9 | Dev_Idle | RO | 1'b0 | **Device is at Idle Interrupt**<br>This bit can be activated by the following two conditions:<br>1. Both A-device and B-device state machines are at the idle state.<br>2. The SessEnd signal is set to high. |
| 8 | DMA_ERROR | R/W | 1'b0 | **DMA Error Interrupt**<br>The DMA operation cannot be finished normally and an error signal is received.<br>When CPU initiates DMA to fill up or read out FIFO of the device and the DMA controller receives an error response from the system bus, this bit will be set. This bit can only be cleared by firmware.<br>This bit will not be affected by a USB bus reset. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 7 | DMA_CMPLT | R/W | 1'b0 | **DMA Completion Interrupt**<br>The DMA operation has finished normally.<br>When CPU initiates DMA to fill up or read out FIFO of the device, this bit will be set after compiling the mission. This bit can only be cleared by firmware.<br>This bit will not be affected by a USB bus reset. |
| 6 | RX0BYTE_INT | R/W | 1'b0 | **Received Zero-length Data Packet Interrupt**<br>The device receives a zero-length data packet from the USB host.<br>When the device receives a zero-length data packet from the USB host, this bit will be set. The firmware may further check the 150H register to determine which endpoint receives a zero-length data packet from the USB host. When the interrupt occurs, the device will NAK the next OUT transaction to the same endpoint until the corresponding bit (In 0x150) is cleared by firmware.<br>This bit will not be affected by a USB bus reset. |
| 5 | TX0BYTE_INT | R/W | 1'b0 | **Transferred Zero-length Data Packet Interrupt**<br>The device returns a zero-length data packet to the USB host.<br>This bit will be set under the following two conditions:<br>• When the USB host issues an IN transaction to an isochronous endpoint and the device is not ready to return data, the device will transfer a zero-length data packet to the USB host. In such case, this bit will be set.<br>• When the TX0BYTE_IEPx bit is set after the endpoint data in FIFO are transferred, the FOTG210 will return a zero-length data packet to the next IN transaction of the same endpoint.<br>The firmware may further check the 154H register to determine which endpoint returns a zero-length data packet to the USB host. After AP serviced the interrupt request, this bit must be cleared by firmware.<br>This bit will not be affected by a USB bus reset. |
| 4 | ISO_SEQ_ABORT_INT | R/W | 1'b0 | **ISO Sequential Abort Interrupt**<br>High bandwidth isochronous sequential abort<br>When the device detects an incomplete DATA PID sequence during a microframe, this bit will be set. For example, if a device detects an MDATA followed by an SOF, this is taken as sequential abort. The firmware should further check register 154H to determine which endpoint receives an isochronous sequential abort. After AP serviced the interrupt request, this bit must be cleared by firmware.<br>This bit is not affected by a USB bus reset. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 3 | ISO_SEQ_ERR_INT | R/W | 1'b0 | **ISO Sequential Error Interrupt**<br>High bandwidth isochronous sequential error<br>When a device detects a DATA PID sequence error of an isochronous transaction in the high bandwidth, this bit will be set. Any out-of-order sequence will be a sequence error. The firmware should further check the 154H register to determine which endpoint receives an isochronous sequential error. After AP serviced the interrupt request, this bit must be cleared by firmware.<br>This bit will not be affected by a USB bus reset. |
| 2 | RESM_INT | R/W | 1'b0 | **Resume Interrupt**<br>Resume-state-change interrupt bit.<br>When a device detects a resume event from host, this bit will be set. After AP services an interrupt request, this bit must be cleared by firmware. When a USB bus reset occurs, this bit will also be cleared. |
| 1 | SUSP_INT | R/W | 1'b0<br>1'b0(U) | **Suspend Interrupt**<br>Suspend-state-change interrupt bit.<br>When the USB bus remains at the idle state over 3 ms, this bit will be set. This bit must be cleared before the firmware sets the "GOSUSP" bit of the 0C0H register. This bit will be cleared when a USB bus reset or resume occurs. |
| 0 | USBRST_INT | R/W | 1'b0<br>1'b1(U) | **USB Reset Interrupt**<br>Bus reset interrupt bit.<br>When a device detects the USB bus reset from host, this bit will be set. When AP services a interrupt request, this bit must be cleared by firmware. |

### 12.4.1.5.15 Device Receive Zero-Length Data Packet Register (Address = 0x150)

All bits are the status bits that respond to the zero-length data packet received by an endpoint.

**Table 12-36. Device Receive Zero-Length Data Packet Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:8] | Reserved | - | - | - |
| 7 | RX0BYTE_EP8 | R/W | 1'b0 | Endpoint 8 receives a zero-length data packet. |
| 6 | RX0BYTE_EP7 | R/W | 1'b0 | Endpoint 7 receives a zero-length data packet. |
| 5 | RX0BYTE_EP6 | R/W | 1'b0 | Endpoint 6 receives a zero-length data packet. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4 | RX0BYTE_EP5 | R/W | 1'b0 | Endpoint 5 receives a zero-length data packet. |
| 3 | RX0BYTE_EP4 | R/W | 1'b0 | Endpoint 4 receives a zero-length data packet. |
| 2 | RX0BYTE_EP3 | R/W | 1'b0 | Endpoint 3 receives a zero-length data packet. |
| 1 | RX0BYTE_EP2 | R/W | 1'b0 | Endpoint 2 receives a zero-length data packet. |
| 0 | RX0BYTE_EP1 | R/W | 1'b0 | Endpoint 1 receives a zero-length data packet. |

#### 12.4.1.5.16 Device Transfer Zero-Length Data Packet Register (Address = 0x154)

All bits are the status bits that response to the zero-length data packet sent by an endpoint.

**Table 12-37. Device Transfer Zero-Length Data Packet Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:8] | Reserved | - | - | - |
| 7 | TX0BYTE_EP8 | R/W | 1'b0 | Endpoint 8 transfers zero-length data packet. |
| 6 | TX0BYTE_EP7 | R/W | 1'b0 | Endpoint 7 transfers zero-length data packet. |
| 5 | TX0BYTE_EP6 | R/W | 1'b0 | Endpoint 6 transfers zero-length data packet. |
| 4 | TX0BYTE_EP5 | R/W | 1'b0 | Endpoint 5 transfers zero-length data packet. |
| 3 | TX0BYTE_EP4 | R/W | 1'b0 | Endpoint 4 transfers zero-length data packet. |
| 2 | TX0BYTE_EP3 | R/W | 1'b0 | Endpoint 3 transfers zero-length data packet. |
| 1 | TX0BYTE_EP2 | R/W | 1'b0 | Endpoint 2 transfers zero-length data packet. |
| 0 | TX0BYTE_EP1 | R/W | 1'b0 | Endpoint 1 transfers zero-length data packet. |

#### 12.4.1.5.17 Device Isochronous Sequential Error/Abort Register (Address = 0x158)

All bits are the status bits that response to the sequential error/abort happened in an ISO endpoint.

**Table 12-38. Device Isochronous Sequential Error/Abort Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:24] | Reserved | - | - | - |
| 23 | ISO_SEQ_ERR_EP8 | R/W | 1'b0 | Endpoint 8 encounters an isochronous sequential error. |
| 22 | ISO_SEQ_ERR_EP7 | R/W | 1'b0 | Endpoint 7 encounters an isochronous sequential error. |
| 21 | ISO_SEQ_ERR_EP6 | R/W | 1'b0 | Endpoint 6 encounters an isochronous sequential error. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 20 | ISO_SEQ_ERR_EP5 | R/W | 1'b0 | Endpoint 5 encounters an isochronous sequential error. |
| 19 | ISO_SEQ_ERR_EP4 | R/W | 1'b0 | Endpoint 4 encounters an isochronous sequential error. |
| 18 | ISO_SEQ_ERR_EP3 | R/W | 1'b0 | Endpoint 3 encounters an isochronous sequential error. |
| 17 | ISO_SEQ_ERR_EP2 | R/W | 1'b0 | Endpoint 2 encounters an isochronous sequential error. |
| 16 | ISO_SEQ_ERR_EP1 | R/W | 1'b0 | Endpoint 1 encounters an isochronous sequential error. |
| [15:8] | Reserved | - | - | - |
| 7 | ISO_ABT_ERR_EP8 | R/W | 1'b0 | Endpoint 8 encounters an isochronous sequential abort. |
| 6 | ISO_ABT_ERR_EP7 | R/W | 1'b0 | Endpoint 7 encounters an isochronous sequential abort. |
| 5 | ISO_ABT_ERR_EP6 | R/W | 1'b0 | Endpoint 6 encounters an isochronous sequential abort. |
| 4 | ISO_ABT_ERR_EP5 | R/W | 1'b0 | Endpoint 5 encounters an isochronous sequential abort. |
| 3 | ISO_ABT_ERR_EP4 | R/W | 1'b0 | Endpoint 4 encounters an isochronous sequential abort. |
| 2 | ISO_ABT_ERR_EP3 | R/W | 1'b0 | Endpoint 3 encounters an isochronous sequential abort. |
| 1 | ISO_ABT_ERR_EP2 | R/W | 1'b0 | Endpoint 2 encounters an isochronous sequential abort. |
| 0 | ISO_ABT_ERR_EP1 | R/W | 1'b0 | Endpoint 1 encounters an isochronous sequential abort. |

#### 12.4.1.5.18 Device IN Endpoint x MaxPacketSize Register (One per Endpoint, x = 1 ~ 8) (Address = 0x160 + 4(x - 1))

**Table 12-39. Device IN Endpoint x MaxPacketSize Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:16] | Reserved | - | - | - |
| 15 | TX0BYTE_IEPx | R/W | 1'b0 | **Transfer a Zero-length Data Packet from Endpoint x to USB Host**<br>This bit should be set after the last packet of a transaction is sent to FIFO. After the endpoint data in FIFO are transferred, the device will return a zero-length data packet to the next IN transaction to the same endpoint.<br>AP should not send the next packet to the same endpoint until TX0BYTE_INT for the endpoint occurs.<br>This bit is cleared by hardware automatically when TX0BYTE_INT occurs. |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [14:13] | TX_NUM_HBW_IEPx | R/W | 2'b0 | **Transaction Number of High Bandwidth Endpoint x**<br>TX_NUM_HBW[1:0] (Only valid for Isochronous transfer)<br>0 0, 0 1: This setting indicates that the endpoint x is with the non-high bandwidth<br>1 0: Two transactions per microframe<br>1 1: Three transactions per microframe |
| 12 | RSTG_IEPx | R/W | 1'b0 | **Reset Toggle Sequence for IN Endpoint x**<br>The firmware resets the toggle bit of the indexed endpoint x by writing '1' to this bit. This bit is cleared by firmware. |
| 11 | STL_IEPx | R/W | 1'b0<br>1'b0(U) | **Stall IN Endpoint x**<br>The indexed endpoint x will be stalled by writing '1' to this bit. The stalled status of the indexed endpoint x will be cleared by writing '0' to this bit. Before setting this bit, users should check the FIFO empty register (0x120) and make sure that the related FIFO is empty. |
| [10:0] | MAXPS_IEPx | R/W | 11'h200 | **Max Packet Size of IN Endpoint x**<br>The maximum packet size of endpoint x is capable of sending or receiving data of smaller than or equal to this size (Not exceeding the FIFO size). |

### 12.4.1.5.19 Device OUT Endpoint x MaxPacketSize Register (One per Endpoint, x = 1 ~ 8) (Address = 0x180 + 4(x - 1))

**Table 12-40. Device OUT Endpoint x MaxPacketSize Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:13] | Reserved | - | - | - |
| 12 | RSTG_OEPx | R/W | 1'b0 | **Reset Toggle Sequence for OUT Endpoint x**<br>The firmware resets the toggle bit of the indexed endpoint x by writing '1' to this bit. This bit is cleared by firmware. |
| 11 | STL_OEPx | R/W | 1'b0 | **Stall OUT Endpoint x**<br>Setting this bit to '1' will stall endpoint x. The stalled status of the indexed endpoint x will be cleared by writing '0' to this bit. |
| [10:0] | MAXPS_OEPx | R/W | 11'h200 | **Max Packet Size of OUT Endpoint x**<br>The maximum packet size of endpoint x is capable of sending or receiving data of smaller than or equal to this size. |

### 12.4.1.5.20 Device Endpoint 1 ~ 4 Map Register (Address = 0x1A0)

**Table 12-41. Device Endpoint 1 ~ 4 Map Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:30] | Reserved | - | - | - |
| [29:28] | FNO_OEP4 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 4**<br>This field records the physical FIFO number of logical OUT endpoint 4. |
| [27:26] | Reserved | - | - | - |
| [25:24] | FNO_IEP4 | R/W | 2'b11 | **FIFO Number of IN Endpoint 4**<br>This register records the physical FIFO number of logical IN endpoint 4. |
| [23:22] | Reserved | - | - | - |
| [21:20] | FNO_OEP3 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 3**<br>This field records the physical FIFO number of logical OUT endpoint 3. |
| [19:18] | Reserved | - | - | - |
| [17:16] | FNO_IEP3 | R/W | 2'b11 | **FIFO Number of IN Endpoint 3**<br>This field records the physical FIFO number of logical IN endpoint 3. |
| [15:14] | Reserved | - | - | - |
| [13:12] | FNO_OEP2 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 2**<br>This field records the physical FIFO number of logical OUT endpoint 2. |
| [11:10] | Reserved | - | - | - |
| [9:8] | FNO_IEP2 | R/W | 2'b11 | **FIFO Number of IN Endpoint 2**<br>This field records the physical FIFO number of logical IN endpoint 2. |
| [7:6] | Reserved | - | - | - |
| [5:4] | FNO_OEP1 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 1**<br>This field records the physical FIFO number of logical OUT endpoint 1. |
| [3:2] | Reserved | - | - | - |
| [1:0] | FNO_IEP1 | R/W | 2'b11 | **FIFO Number of IN Endpoint 1**<br>This field records the physical FIFO number of logical IN endpoint 1. |

### 12.4.1.5.21 Device Endpoint 5 ~ 8 Map Register (Address = 0x1A4)

**Table 12-42. Device Endpoint 5 ~ 8 Map Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:30] | Reserved | - | - | - |
| [29:28] | FNO_OEP8 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 8**<br>This field records the physical FIFO number of logical OUT endpoint 8. |
| [27:26] | Reserved | - | - | - |
| [25:24] | FNO_IEP8 | R/W | 2'b11 | **FIFO Number of IN Endpoint 8**<br>This field records the physical FIFO number of logical IN endpoint 8. |
| [23:22] | Reserved | - | - | - |
| [21:20] | FNO_OEP7 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 7**<br>This field records the physical FIFO number of logical OUT endpoint 7. |
| [19:18] | Reserved | - | - | - |
| [17:16] | FNO_IEP7 | R/W | 2'b11 | **FIFO Number of IN Endpoint 7**<br>This field records the physical FIFO number of logical IN endpoint 7. |
| [15:14] | Reserved | - | - | - |
| [13:12] | FNO_OEP6 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 6**<br>This field records the physical FIFO number of logical OUT endpoint 6. |
| [11:10] | Reserved | - | - | - |
| [9:8] | FNO_IEP6 | R/W | 2'b11 | **FIFO Number of IN Endpoint 6**<br>This field records the physical FIFO number of logical IN endpoint 6. |
| [7:6] | Reserved | - | - | - |
| [5:4] | FNO_OEP5 | R/W | 2'b11 | **FIFO Number of OUT Endpoint 5**<br>This field records the physical FIFO number of logical OUT endpoint 5. |
| [3:2] | Reserved | - | - | - |
| [1:0] | FNO_IEP5 | R/W | 2'b11 | **FIFO Number of IN Endpoint 5**<br>This field records the physical FIFO number of logical IN endpoint 5. |

### 12.4.1.5.22 Device FIFO Map Register (Address = 0x1A8)

**Table 12-43. Device FIFO Map Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:30] | Reserved | - | - | - |
| [29:28] | Dir_FIFO3 | R/W | 2'b0 | **FIFO 3 Direction**<br>Data transfer direction<br>2'b00: OUT<br>2'b01: IN<br>2'b10: Bidirectional<br>2'b11: Not allowed |
| [27:24] | EPNO_FIFO3 | R/W | 4'b1111 | **Endpoint number of FIFO 3**<br>This field records the physical endpoint number of the physical FIFO 3. |
| [23:22] | Reserved | - | - | - |
| [21:20] | Dir_FIFO2 | R/W | 2'b0 | **FIFO 2 Direction**<br>Data transfer direction<br>2'b00: OUT<br>2'b01: IN<br>2'b10: Bidirectional<br>2'b11: Not allowed |
| [19:16] | EPNO_FIFO2 | R/W | 4'b1111 | **Endpoint number of FIFO 2**<br>This field records the physical endpoint number of the physical FIFO 2. |
| [15:14] | Reserved | - | - | - |
| [13:12] | Dir_FIFO1 | R/W | 2'b0 | **FIFO 1 Direction**<br>Data transfer direction<br>2'b00: Out<br>2'b01: In<br>2'b10: Bidirectional<br>2'b11: Not allowed |
| [11:8] | EPNO_FIFO1 | R/W | 4'b1111 | **Endpoint number of FIFO 1**<br>This field records the physical endpoint number of the physical FIFO 1. |
| [7:6] | Reserved | - | - | - |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [5:4] | Dir_FIFO0 | R/W | 2'b0 | **FIFO 0 Direction**<br>Data transfer direction<br>2'b00: Out<br>2'b01: In<br>2'b10: Bidirectional<br>2'b11: Not allowed |
| [3:0] | EPNO_FIFO0 | R/W | 4'b1111 | **Endpoint number of FIFO 0**<br>This field records the physical endpoint number of the physical FIFO 0. |

### 12.4.1.5.23 Device FIFO Configuration (Address = 0x1AC)

**Table 12-44. Device FIFO Configuration**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:30] | Reserved | - | - | - |
| 29 | EN_F3 | R/W | 1'b0 | **Enable FIFO 3**<br>Setting this bit to '1' to indicate that the FIFO is enabled. |
| 28 | BLKSZ_F3 | R/W | 1'b0 | **Block Size of FIFO 3**<br>BLKSIZE_F3 = 0: The maximum packet size of the transferred packets is smaller than or equal to 512 bytes.<br>BLKSIZE_F3 = 1: The maximum packet size of the transferred packets is smaller than or equal to 1024 bytes but greater than 512 bytes. |
| [27:26] | BLKNO_F3 | R/W | 2'b0 | **Block Number of FIFO 3**<br>BLKNUM_F3 = 00 : Single block<br>BLKNUM_F3 = 01 : Double blocks<br>BLKNUM_F3 = 10 : Triple blocks<br>BLKNUM_F3 = 11 : Reserved |
| [25:24] | BLK_TYP_F3 | R/W | 2'b0 | **Transfer Type of FIFO 3**<br>This field indicates the transfer type used for FIFOx transfer.<br>TYP_F3 = 00 : Reserved<br>TYP_F3 = 01 : Isochronous type<br>TYP_F3 = 10 : Bulk type<br>TYP_F3 = 11 : Interrupt type |
| [23:22] | Reserved | - | - | - |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 21 | EN_F2 | R/W | 1'b0 | **Enable FIFO 2**<br>Setting this bit to '1' to indicate that the FIFO is enabled. |
| 20 | BLKSZ_F2 | R/W | 1'b0 | **Block Size of FIFO 2**<br>BLKSIZE_F2 = 0: The maximum packet size of the transferred packets is smaller than or equal to 512 bytes.<br>BLKSIZE_F2 = 1: The maximum packet size of the transferred packets is smaller than or equal to 1024 bytes but greater than 512 bytes. |
| [19:18] | BLKNO_F2 | R/W | 2'b0 | **Block Number of FIFO 2**<br>BLKNUM_F2 = 00: Single block<br>BLKNUM_F2 = 01: Double blocks<br>BLKNUM_F2 = 10: Triple blocks<br>BLKNUM_F2 = 11: Reserved |
| [17:16] | BLK_TYP_F2 | R/W | 2'b0 | **Transfer Type of FIFO 2**<br>This register indicates the transfer type used for FIFO 2 transfer.<br>TYP_F2 = 00: Reserved<br>TYP_F2 = 01: Isochronous type<br>TYP_F2 = 10: Bulk type<br>TYP_F2 = 11: Interrupt type |
| [15:14] | Reserved | - | - | - |
| 13 | EN_F1 | R/W | 1'b0 | **Enable FIFO 1**<br>A '1' indicates that the FIFO is enabled. |
| 12 | BLKSZ_F1 | R/W | 1'b0 | **Block Size of FIFO 1**<br>BLKSIZE_F1 = 0: The maximum packet size of the transferred packets is smaller than or equal to 512 bytes.<br>BLKSIZE_F1 = 1: The maximum packet size of the transferred packets is smaller than or equal to 1024 bytes but greater than 512 bytes. |
| [11:10] | BLKNO_F1 | R/W | 2'b0 | **Block Number of FIFO 1**<br>BLKNUM_F1 = 00: Single block<br>BLKNUM_F1 = 01: Double blocks<br>BLKNUM_F1 = 10: Triple blocks<br>BLKNUM_F1 = 11: Reserved |
| [9:8] | BLK_TYP_F1 | R/W | 2'b0 | **Transfer Type of FIFO 1**<br>the transfer type used for FIFO 1 transfer.<br>TYP_F1 = 00: Reserved<br>TYP_F1 = 01: Isochronous type<br>TYP_F1 = 10: Bulk type<br>TYP_F1 = 11: Interrupt type |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [7:6] | Reserved | - | - | - |
| 5 | EN_F0 | R/W | 1'b0 | **Enable FIFO 0**<br>A '1' indicates that the FIFO is enabled. |
| 4 | BLKSZ_F0 | R/W | 1'b0 | **Block Size of FIFO 0**<br>BLKSIZE_F0 = 0: The maximum packet size of the transferred packets is smaller than or equal to 512 bytes.<br>BLKSIZE_F0 = 1: The maximum packet size of the transferred packets is smaller than or equal to 1024 bytes but greater than 512 bytes. |
| [3:2] | BLKNO_F0 | R/W | 2'b0 | **Block Number of FIFO 0**<br>BLKNUM_F0 = 00: Single block<br>BLKNUM_F0 = 01: Double blocks<br>BLKNUM_F0 = 10: Triple blocks<br>BLKNUM_F0 = 11: Reserved |
| [1:0] | BLK_TYP_F0 | R/W | 2'b0 | **Transfer Type of FIFO 0**<br>This field indicates the transfer type used for FIFO 0 transfer.<br>TYP_F0 = 00: Reserved<br>TYP_F0 = 01: Isochronous type<br>TYP_F0 = 10: Bulk type<br>TYP_F0 = 11: Interrupt type |

### 12.4.1.5.24 Device FIFO x Instruction and Byte Count Register (One per FIFO, x = 0 ~ 3) (Address = 0x1B0 + 4 * x)

**Table 12-45. Device FIFO x Instruction and Byte Count Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:13] | Reserved | - | - | - |
| 12 | FFRST | R/W | 1'b0 | **FIFO x reset**<br>FIFO is reset by firmware through setting this bit. This bit is cleared automatically. |
| 11 | Reserved | - | - | - |
| [10:0] | BC_Fx | RO | 11'b0 | **OUT FIFO x Byte Count**<br>BC_Fx[10:0] indicates the byte number of data stored in FIFO for OUT EPX. |

### 12.4.1.5.25 Device DMA Target FIFO Number Register (Address = 0x1C0)

Please note that only one of the ACC_X bits can be set at the same time.

**Table 12-46. Device DMA Target FIFO Number Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:5] | Reserved | - | - | - |
| 4 | ACC_CXF | R/W | 1'b0 | **Accessing Control Transfer FIFO**<br>When this bit is set to '1', the DMA target FIFO is the control transfer FIFO. |
| 3 | ACC_F3 | R/W | 1'b0 | **Accessing FIFO 3**<br>When this bit is set to '1', the DMA target FIFO is FIFO 3. |
| 2 | ACC_F2 | R/W | 1'b0 | **Accessing FIFO 2**<br>When this bit is set to '1', the DMA target FIFO is FIFO 2. |
| 1 | ACC_F1 | R/W | 1'b0 | **Accessing FIFO 1**<br>When this bit is set to '1', the DMA target FIFO is FIFO 1. |
| 0 | ACC_F0 | R/W | 1'b0 | **Accessing FIFO 0**<br>When this bit is set to '1', the DMA target FIFO is FIFO 0. |

### 12.4.1.5.26 Device DMA Controller Parameter Setting 1 Register (Address = 0x1C8)

**Table 12-47. Device DMA Controller Parameter Setting 1 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 31 | DevPhy_Suspend | R/W- | 1'b0- | **Device Transceiver Suspend Mode**<br>Active high of this bit will place the transceiver in the suspend mode to draw the minimal power from the power supplies. This bit can only be used in the device mode. This bit is in the system clock domain, not the UCLK clock domain. |
| [30:25] | Reserved | - | - | - |
| [24:8] | DMA_LEN | R/W | 17'h00000 | **DMA Length**<br>The total bytes that the DMA controller can move<br>The unit is byte. The maximum length is (128 kB – 1) and must not be configured to '0'. When handling the control transfer, the maximum length must not exceed 64. |
| [7:5] | Reserved | - | - | - |

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| 4 | CLRFIFO_DMAABORT | R/W | 1'b0 | **Clear FIFO when DMA_ABORT**<br>When this bit is set to 1'b1 and combined with the DMA_ABORT bit, the contents in FIFO will be cleared after the DMA abort is complete. If users need to abort DMA and clear the contents in FIFO, together with DMA_ABORT, this bit must be set to 1'b1. The contents in FIFO will not be cleared if this bit is cleared to 1'b0. |
| 3 | DMA_ABORT | R/W | 1'b0 | **DMA Abort**<br>Force the DMA abort during active DMA<br>This bit is set to 1'b1 to stop the DMA data movement and will be cleared when DMA is stopped. Please note that this bit will be valid only in the device mode and DMA_START and DMA_ABORT cannot be set simultaneously. By doing this, unexpected outcomes will occur. Setting this bit to 1'b1 while the DMA_START bit is 1'b0 will take no effect. |
| 2 | DMA_IO | R/W | 1'b0 | **DMA I/O to I/O**<br>The DMA controller is force not to toggle address. This bit is set when the DMA target is not sent to the system memory but to the I/O device. If this bit is set to '1', "DMA_LEN" must be an integer multiple of DWORD (4 bytes) and "DMA_MADDR" must align to the boundary of DWORD (4 bytes). |
| 1 | DMA_TYPE | R/W | 1'b0 | **DMA Type**<br>The transfer type of data moving<br>0: FIFO to Memory<br>1: Memory to FIFO |
| 0 | DMA_START | R/W | 1'b0 | **DMA Start**<br>This bit informs the DMA controller to initiate a DMA transfer. Setting this bit will start a transfer and then be cleared when the DMA operation is completed. Please note that this bit cannot be cleared by software; it can only be cleared by hardware in the case of either DMA completion or DMA error. Please note that DMA_LEN must be configured before DMA_START is set. |

### 12.4.1.5.27 Device DMA Controller Parameter Setting 2 Register (Address = 0x1CC)

**Table 12-48. Device DMA Controller Parameter Setting 2 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:0] | DMA_MADDR | R/W | 32'h0000 | **DMA Memory Address**<br>The starting address of memory to request a DMA transfer |

### 12.4.1.5.28 Device DMA Controller Parameter Setting 3 Register (Address = 0x1D0)

**Table 12-49. Device DMA Controller Parameter Setting 3 Register**

| Bit | Name | Type | Default Value | Description |
|---|---|---|---|---|
| [31:0] | SETUP_CMD_RPORT | RO | - | **Control Transfer Setup Command Read Port**<br>CPU reads the 8-byte setup command from this port instead of programming DMA to move data.<br>CPU must read one word each time when it reads this port (A half-word or byte read is not allowed). Before reading this port, the DMA target FIFO (0x1C0) must be set to control the transfer FIFO even though DMA is not used to move data.<br>When CPU reads this port, the internal FIFO pointer will be increased. In the case, the pointer increases erroneously and users must not set the target FIFO to the control transfer FIFO while using ICE to scan all ports. |

# Chapter 13

# H.264 Encoder

This chapter contains the following sections:


- 13.1 General Description
- 13.2 Features
- 13.3 Block Diagram

## 13.1 General Description

H.264 is a hardware video encoder for the baseline profile of the MPEG4 AVC/JVT/H.264 video coding standard. This encoder is designed to compress a sequence of YCbCr 4:2:0 pictures to a compressed video bitstream. The supported resolution is ranging from 128 x 80 to 1920 x 1088 with steps in 16 units. For better quality, this encoder supports variable block sizes in the inter-prediction. The motion vector is quarter-pel accurate for each block type and the search range is ±24 in the horizontal direction, and ±16 in the vertical direction. The clock rate of encoding a SDTV sequence in real-time is under 54 MHz.

## 13.2 Features

- MPEG4 AVC/JVT/H.264 (ISO/IEC 14496-10) video coding standards
- Baseline profile to level 3.1
- I and P frame encodings
- CBR and VBR (Rate control by firmware)
- Programmable the in-loop filter parameters
- Programmable Chroma QP index offset parameter

## 13.3 Block Diagram

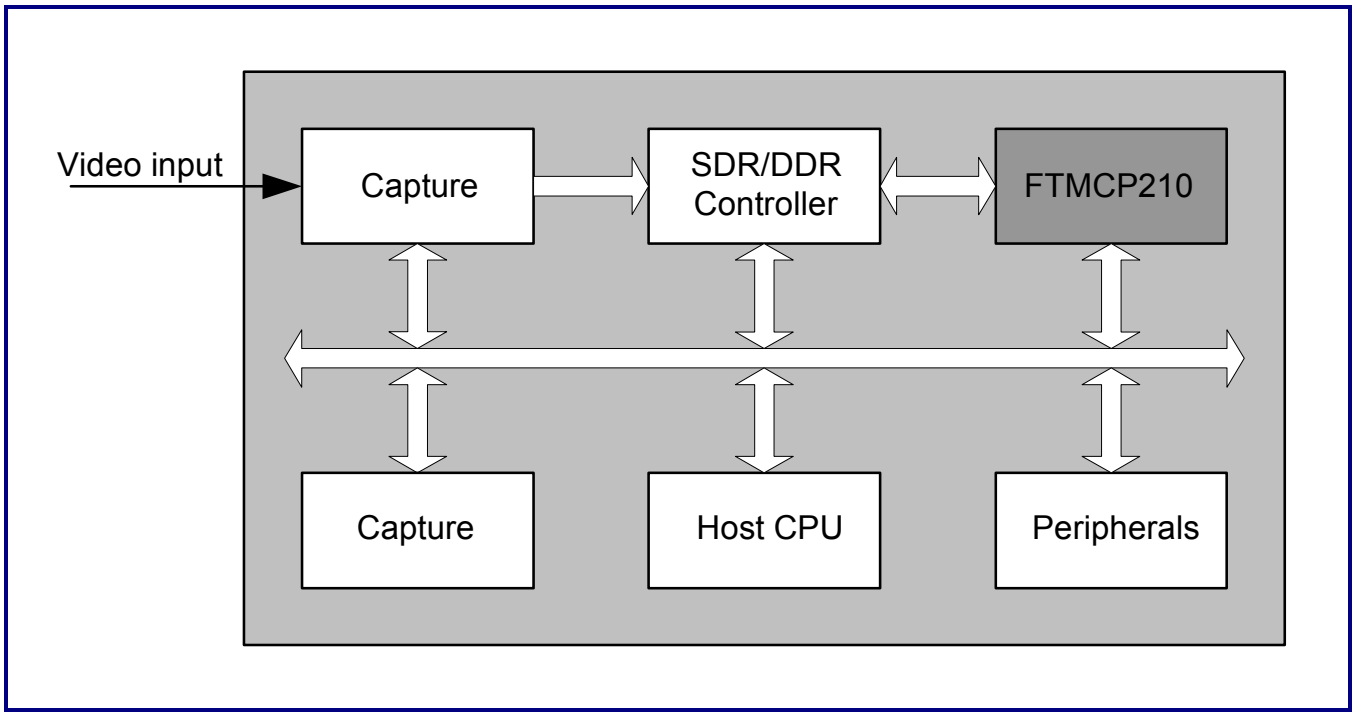



**Figure 13-1. Block Diagram of FTMCP210**

### 13.3.1 Video Input Format

The H.264 encoder accepts the YCbCr 420 video format as the input. Each component of YCbCr is arranged in separated memory spaces. The pixel data in each macroblock are arranged in consecutive addresses. The macroblocks in each frame are arranged in roster scan sequence.

### 13.3.2 AHB Host Interface

The AHB host interface is used to program the FTMCP210 engine by the host CPU.

### 13.3.3 DMA

The built-in DMA engine is used to transfer data between the system memory and the local memory through the AMBA® AHB interface. These data include the current frame, reference frame, temporal prediction result, temporal in-loop filter result, temporal marcoblock information, and encoded bitstream.

### 13.3.4 Motion Estimation

The Motion Estimation (ME) engine is used to find out the best motion vector and the best macroblock partition that have the best block match. A modified full-search algorithm is used to scan all the defined search range in full-pel precision. A quarter-pel precision refinement will be performed for the best full-pel motion vector of each partition block in a marcoblock.

### 13.3.5 Intra Prediction

The intra prediction engine is used to find out a best prediction mode to encode the marcoblock in the intra mode and its cost value.

### 13.3.6 Entropy Coding

The entropy coding unit supports the H.264 CAVLC coding mode.

### 13.3.7 TQ and IQIT

This module is used to generate quantized transform coefficient for entropy coding. It also performs the inverse quantization and inverse transform to generate reconstructed data for deblocking engine.

### 13.3.8 Deblocking Filter

This filter is applied to each inverse transformed macroblock for reducing the blocking distortion. This filter is applied to the vertical and horizontal edges of the 4x4 blocks in a macroblock with five pre-defined strength modes.

### 13.3.9 System Controller

The local memory controller arbitrates the local memory access requests from CPU, DMA, and the H.264 encoder. The local memory controller supports several addressing modes to achieve optimum utilization of the local memory bandwidth.

For detailed technical and programming information about the H.264 encoder, please refer to the FTMCP210 Data Sheet.

# Chapter 14

# MPEG/JPEG Engine

This chapter contains the following sections:


- 14.1 General Description
- 14.2 Features
- 14.3 Block Diagram

## 14.1 General Description

FTMCP100 is an AHB-based codec which is capable of accelerating the multimedia image process and the video related applications, such as MPEG4 and JPEG. This codec includes the hardware engines to accelerate the computation-intensive tasks, such as the motion estimation, DCT/IDCT, quantization/ inverse quantization, and motion compensation. FTMCP100 is controlled by CPU through the AHB slave interface. By initializing the control registers of the codec, the motion estimation calculation task for an entire 16 x 16 or 8 x 8 block is done automatically by FTMCP100. The DCT/IDCT, quantization/inverse quantization, AC/DC prediction, zigzag scan, and VLC/VLD calculation tasks are also done automatically for a macro-block by FTMCP100. Thus, CPU is released from the timing-critical tasks in video encoding. To improve the quality of the decoded output, a deblocking post-filter is included, which improves the subjective and objective qualities, especially when the quantization step size is large. This optional feature will be turned on or off by using the hardware configuration or through the software programming. The standard-cell-based approach allows users to quickly integrate FTMCP100 into the user SoC designs. FTMCP100 includes additional features, which are described in the feature list.

The DMA controller performs the task of transferring data between the system memory and the local memory of the MPEG4 codec. The DMA controller includes one AHB master interface and one AHB slave interface. The AHB master interface allows the DMA controller to access data by the AHB bus. The AHB slave interface is used to program the control register of the DMA controller from the AHB bus. Figure 14-1 shows a typical basic system configuration, which incorporates a 32-bit CPU, an FTMCP100, a capture/display interface, and a system memory controller for the MPEG video codec.

## 14.2 Features

**FTMCP100**

**Functions:**

- Supports AHB 2.0 interface
- Compliant with MPEG-4 (ISO/IEC 14496-2) simple profile L0 ~ L3 standards, including resolutions of Sub QCIF, QCIF, CIF, VGA, 4CIF, and D1 at 30 fps with steps of 16 units
- Compliant with JPEG (ISO/IEC 10918-1) base-line standard
- Includes hardware engine for motion estimation/compensation, DCT/IDCT, quantization/ inverse quantization, AC/DC prediction, and variable length coding/decoding

- Supports local memory controller for local memory shared by DMA master and other FTMCP100 blocks
- Supports DMA controller for data transfers between system memory and local memory
- Supports automatic power-down mechanism
- Motion estimation search range: -16 ~ +15.5 (Optionally -32 ~ +31.5) with half-pixel accuracy
- Supports 4 MV and unrestricted MV
- Supports constant bit rate and variable bit rate
- Error resilient tools: Encoder supports resynchronization marker and header extension code; decoder supports resynchronization marker, header extension code, data partition and RVLC
- Supports short video header (H.263 baseline)
- Supports H.263/MPEG/JPEG quantization methods
- JPEG supported features:
  - Supports 4 user-defined Huffman tables (2AC and 2DC)
  - Supports 4 programmable quantization tables
  - Supports interleave and non-interleave scans
  - Supports YUV 4:4:4, 4:2:2, and 4:2:0 formats
  - Supports image size of up to 64k × 64k
- Supports full-duplex operation (e.g. video phone and video conference) by software switching encoding and task decoding on the same hardware
- Supports embedded RISC to minimize loading of host CPU (Optional)
- Supports deblocking post-filter to enhance decoded output quality (Optional)

**Performance:**

- Supports MPEG4 simple profile encoding up to D1 at 30 fps at codec clock speed under 72 MHz
- Power consumption of less than 1.2 mW/MHz for UMC 0.18 μm process
- Supports gate count below 185K gates for UMC 0.18 μm process (250K gates for embedded RISC)

**DMA**

**Function**

- Supports AMBA 2.0 AHB bus protocol
- Supports chain transfer function
- Supports 2D addressing mode for source address and destination address

**Configuration:**

- FIFO size
  - 128/256/512/1024/2048 bytes

## 14.3 Block Diagram

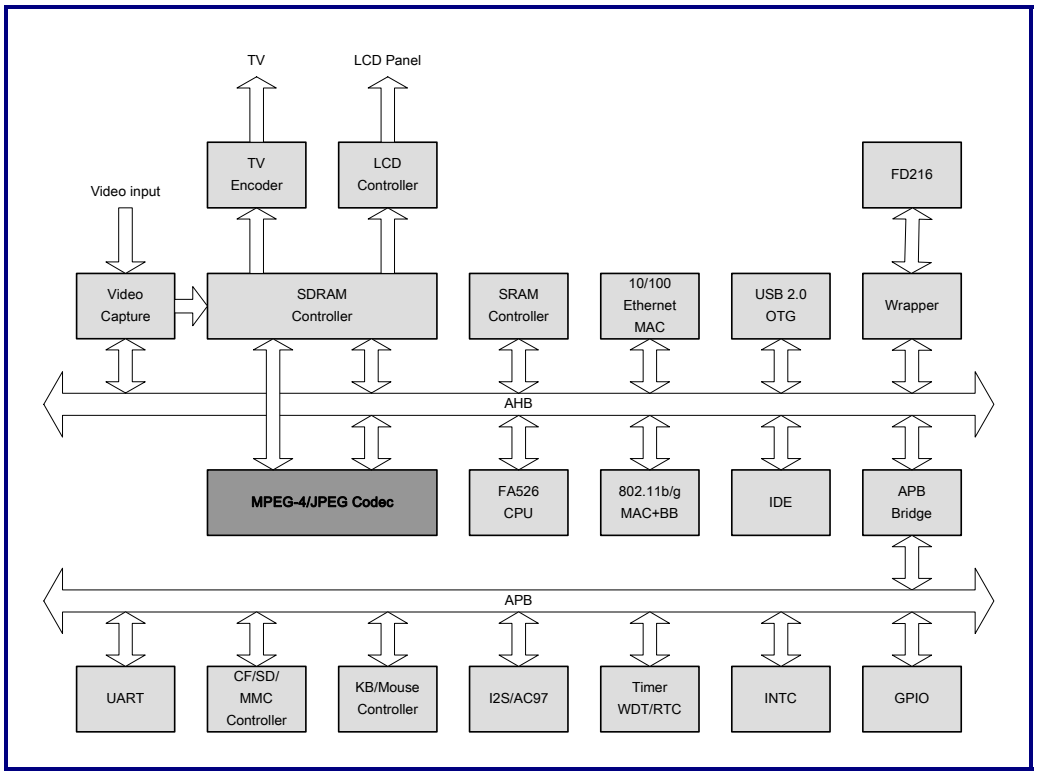



**Figure 14-1. FTMCP100 Block Diagram**

### 14.3.1 AHB Interface

FTMCP100 consists of an AHB master interface and an AHB slave interface. The AHB master interface is used to access the video data from the outside memory to the local memory. The AHB slave interface is used to program the control registers of FTMCP100 and the DMA controller.

### 14.3.2 DMA

The DMA controller transfers data between the system memory and the local memory of FTMCP100. The main function of the DMA controller is to move and translate data into a format that is suitable for processing by other hardware engines.

### 14.3.3 Motion Estimation

The Motion Estimation (ME) unit performs motion estimation for the entire search window based on a fast search algorithm. By reading commands in the local memory, the motion estimation of a macroblock will be completed automatically.

### 14.3.4 DCT/IDCT

The DCT/IDCT unit is responsible for Discrete Cosine Transform (DCT) and Inverse Discrete Cosine Transform (IDCT). The IDCT unit uses the same hardware resources as the DCT unit does. The IDCT results are compliant with the IEEE 1180-1990 specification. The results of the DCT unit are passed to the Quantization unit at the encoding phase and the results of the IDCT unit are passed to the MC unit at the decoding phase.

### 14.3.5 Quantization/Inverse Quantization

The Quantization/Inverse Quantization unit supports the H.263/MPEG/JPEG quantization methods. The results of Quantization are passed to the AC/DC prediction unit at the encoding phase and the inverse results of Quantization are passed to the IDCT unit at the decoding phase.

### 14.3.6 AC/DC Prediction

The AC/DC prediction unit supports both the MPEG-4 AC/DC prediction method and the JPEG DC prediction method. The results of the AC/DC prediction are passed to the Zigzag Scan unit at the encoding phase and the results of the inverse AC/DC prediction are passed to the Inverse Quantization unit at the decoding phase.

### 14.3.7 Zigzag Scan

The Zigzag Scan unit supports all MPEG-4 scan methods and JPEG zigzag method. The results of the Zigzag (Run, level) pairs are passed to the VLC unit during the encoding phase and the results of the inverse zigzag are passed to the AC/DC prediction unit at the decoding phase.

### 14.3.8 Variable Length Coding/Decoding

The Variable Length Coding/Decoding (VLC/VLD) unit supports the MPEG-4 fixed variable length codes and JPEG user-defined Huffman codes. The results of VLC are final compressed bitstreams at the encoding phase and are passed to the Zigzag unit at the decoding phase.

### 14.3.9 Motion Compensation

The Motion Compensation (MC) unit is an engine responsible for motion compensation task. At the encoding phase, this unit subtracts the interpolated block from the current block and sends the residual block to DCT. At the decoding phase, this unit adds the interpolated block to the IDCT output block to get the reconstructed block.

### 14.3.10 Local Memory Controller

The local memory controller arbitrates the local memory access requests from the CPU, DMA, and FTMCP100 codec engines. A multi-bank memory architecture is used for the local memory to improve the memory bandwidth. CPU has the highest access priority, followed by DMA and then codec. The local memory controller supports the following addressing modes for codec operations:

- Transfer a linear block or 2D block from the local memory to codec
- Transfer the codec output to a linear or 2D block in the local memory
- Transfer a 2D block to the ME unit in the circular buffer addressing mode

# Chapter 15

# LCD Controller

This chapter contains the following sections:


- 15.1 General Description
- 15.2 Features
- 15.3 Block Diagram
- 15.4 Function Description
- 15.5 Programming Model

## 15.1 General Description

The LCD controller acquires the video data from the frame buffer and output to provide all necessary control signals for various TFT LCD monitors or TV encoders.

## 15.2 Features

- **TFT panel interface**
  - Supports pixel clock rate of up to 150 MHz
  - Swaps between RGB and BGR
  - Programmable resolution of up to 1024x768
  - Programmable polarity/duration for output enable, vertical sync., horizontal sync., and pixel clock
  - Data/Synchronization on/off controls
  - Virtual screen
- **TV interface**
  - ITU-R BT. 656 output (SDTV)
  - ITU-R BT. 1120 output (HDTV: 1080i, 1080p)
  - Supports VBI
- **Image input format**
  - RGB 12 (4:4:4)/16 (5:6:5)/15 (5:5:5)/24 (8:8:8) bits per pixel
  - Palette (8-bit, 4-bit, 2-bit, and 1-bit pixels)
  - YcbCr422 (16-bit per pixel)
  - YcbCr420 (Separate memory locations)
  - Chooses the input format (Except for YCbCr420 format) as interlace or progress arrangement for TV output
  - Little-endian, big-endian, and WinCE
- **Palette**
  - 256 entries 16-bit RGB color palette RAM

- **Interrupt control**
  - Master bus error
  - Frame status
  - FIFO under-run
  - Memory base update
- **Simple OSD**
  - RAM-based programmable size: 12x16
  - Font variety of up to 256
  - Maximal window fonts: 512
  - Supports flexible dimensions for both horizontal and vertical directions
  - Programmable window position
  - Supports up-scaling function with factors of 1, 2, 3, and 4
  - Transparency types: 25%, 50%, 75%, and 100%
  - 4-entry and 8-bit color palette for foreground
  - 3-entry and 8-bit color palette for background (One entry reserved for transparency)
- **Scaler** (Does not support the TV interlace input or interlace output)
  - Down-scaling ration ranging from $\frac{1}{256} \times \frac{1}{256}$ to 1x1
  - Arbitrary ration between 1x1 and 2x2 for up-scaling stage
  - Three interpolation modes for up-scaling: nearly bilinear interpolation, threshold nearly bilinear interpolation, and most neighborhood interpolation

## 15.3 Block Diagram

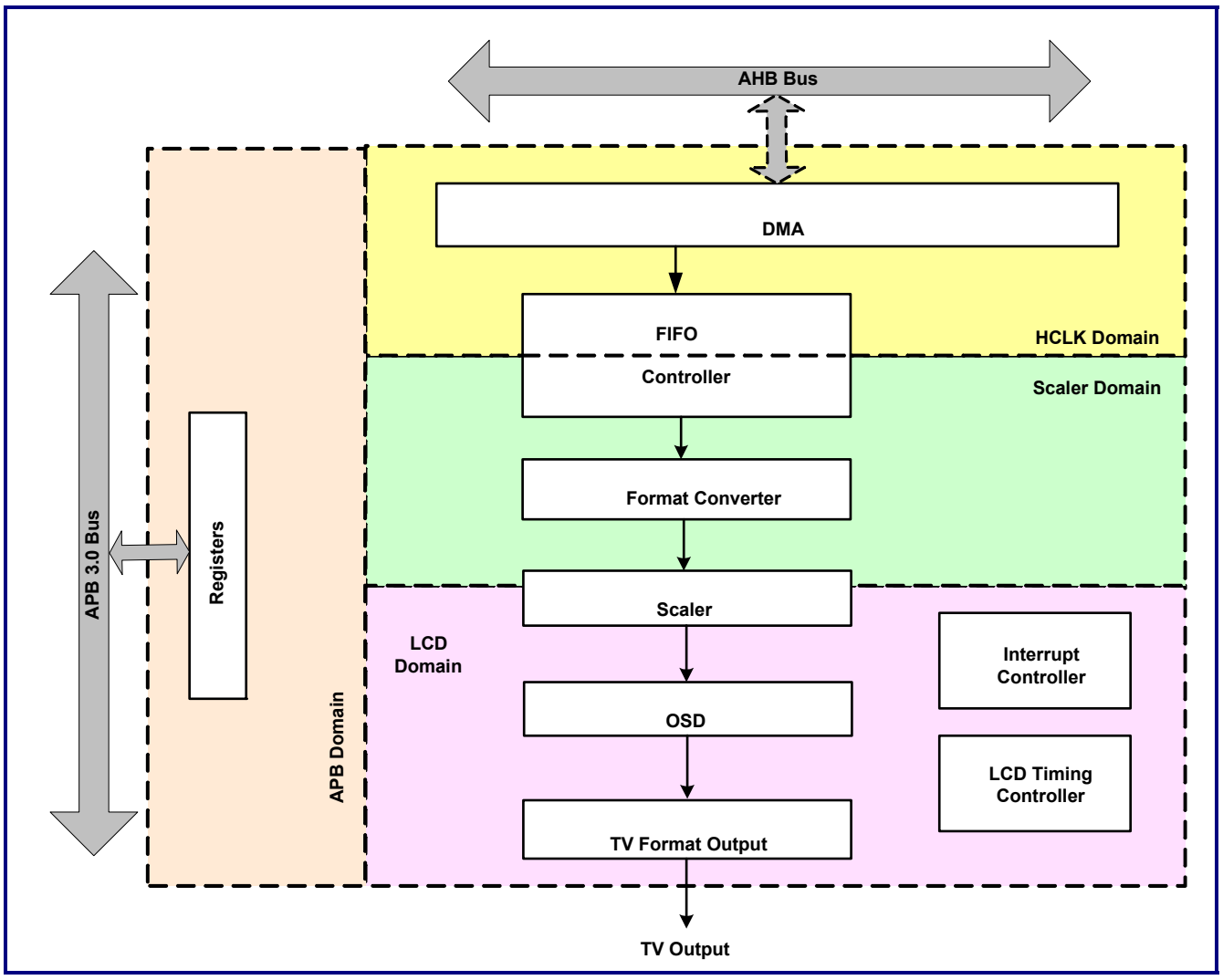



**Figure 15-1. Block Diagram**

## 15.4 Function Description

### 15.4.1 FIFO Controllers and FIFO

Data fetched from an external memory by the DMA master interface are placed into FIFO. Only one controller/FIFO is required, except for the enabled YCbCr420 mode. Three channels, Y, Cb, and Cr, are required because each channel belongs to a particular memory location. The FIFO width is set to 32 bits with configurable depth. The FIFO input port is connected to the AMBA AHB master output and the output port is connected to the unpack controller of the pixel data. Under the condition that the bus bandwidth is unable to support the pixel rate, the unpredictable image distortions will occur. The FIFO controller provides an interrupt signal, "under-run interrupt", to inform the micro-controller for solving the bus congestion problem.

### 15.4.2 Pixel Data Unpack

Data stored in FIFO are 32-bit wide and are packed by the 24-bit, 16-bit, 8-bit, 4-bit, 2-bit, or 1-bit pixel format, depending on the setting of the pixel format. When the YCbCr420 or YcbCr422 mode is enabled, the packing format is fixed and cannot be changed. Depending on the operation mode, the pixel data are used to address the color palette RAM or form the raw color value, and are be directly applied to the LCD panel. Figure 15-2, Figure 15-3, and Figure 15-4 illustrate the format of the packed data. The LCD controller provides 1-bit, 2-bit, 4-bit, 8-bit, 16-bit, and 24-bit BPP (bpp) formats in big-endian byte and big-endian pixel (BBBP), little-endian byte and big-endian pixel (LBBP), or little-endian byte and little-endian pixel (LBLP).

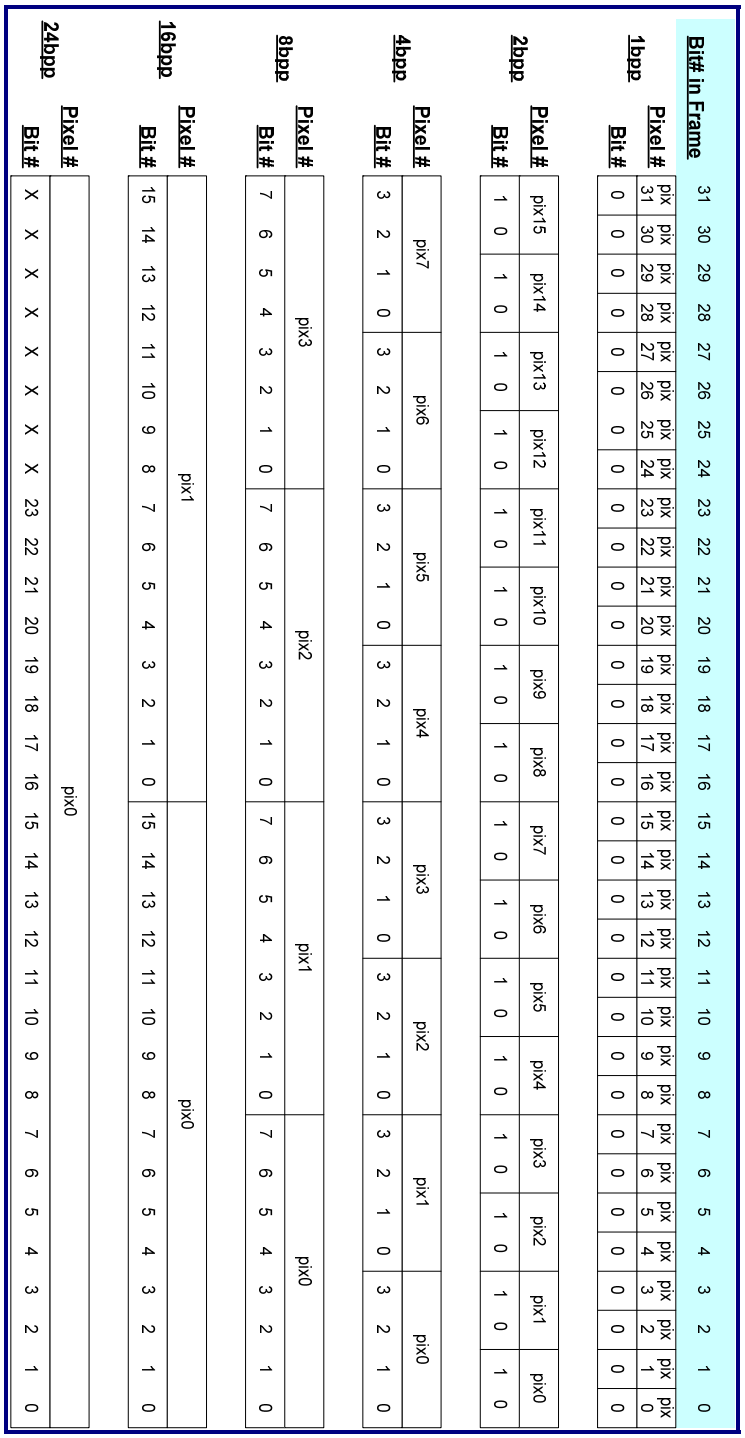



**Figure 15-2. Little-endian Byte and Little-endian Pixel (LBLP)**

| Bit# in Frame | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1bpp** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **Pixel #** | pix 0 | pix 1 | pix 2 | pix 3 | pix 4 | pix 5 | pix 6 | pix 7 | pix 8 | pix 9 | pix 10 | pix 11 | pix 12 | pix 13 | pix 14 | pix 15 | pix 16 | pix 17 | pix 18 | pix 19 | pix 20 | pix 21 | pix 22 | pix 23 | pix 24 | pix 25 | pix 26 | pix 27 | pix 28 | pix 29 | pix 30 | pix 31 |
| **Bit #** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| **2bpp** | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Pixel #** | pix0 | pix1 | pix2 | pix3 | pix4 | pix5 | pix6 | pix7 | pix8 | pix9 | pix10 | pix11 | pix12 | pix13 | pix14 | pix15 |
| **Bit #** | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 | 1 0 |

| **4bpp** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Pixel #** | pix0 | pix1 | pix2 | pix3 | pix4 | pix5 | pix6 | pix7 |
| **Bit #** | 3 2 1 0 | 3 2 1 0 | 3 2 1 0 | 3 2 1 0 | 3 2 1 0 | 3 2 1 0 | 3 2 1 0 | 3 2 1 0 |

| **8bpp** | | | | |
|---|---|---|---|---|
| **Pixel #** | pix0 | pix1 | pix2 | pix3 |
| **Bit #** | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 |

| **16bpp** | | |
|---|---|---|
| **Pixel #** | pix0 | pix1 |
| **Bit #** | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |

| **24bpp** | |
|---|---|
| **Pixel #** | pix0 |
| **Bit #** | X X X X X X X X 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 |

**Figure 15-3. Big-endian Byte and Big-endian Pixel (BBBP)**

| Bit# in Frame | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1bpp** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pixel # | pix 0 | pix 1 | pix 2 | pix 3 | pix 4 | pix 5 | pix 6 | pix 7 | pix 8 | pix 9 | pix 10 | pix 11 | pix 12 | pix 13 | pix 14 | pix 15 | pix 16 | pix 17 | pix 18 | pix 19 | pix 20 | pix 21 | pix 22 | pix 23 | pix 24 | pix 25 | pix 26 | pix 27 | pix 28 | pix 29 | pix 30 | pix 31 |
| Bit # | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2bpp** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pixel # | pix12 | | | | pix13 | | | | pix14 | | | | pix15 | | | | pix8 | | | | pix9 | | | | pix10 | | | | pix11 | | | |
| Bit # | 1 | 0 | | | 1 | 0 | | | 1 | 0 | | | 1 | 0 | | | 1 | 0 | | | 1 | 0 | | | 1 | 0 | | | 1 | 0 | | |
| **4bpp** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pixel # | pix6 | | | | pix7 | | | | pix4 | | | | pix5 | | | | pix4 | | | | pix5 | | | | pix6 | | | | pix7 | | | |
| Bit # | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 3 | 2 | 1 | 0 |
| **8bpp** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pixel # | pix3 | | | | | | | | pix2 | | | | | | | | pix1 | | | | | | | | pix0 | | | | | | | |
| Bit # | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **16bpp** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pixel # | pix1 | | | | | | | | | | | | | | | | pix0 | | | | | | | | | | | | | | | |
| Bit # | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **24bpp** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pixel # | pix0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bit # | X | X | X | X | X | X | X | X | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

**Figure 15-4. Little-endian Byte and Big-endian Pixel (LBBP)**

### 15.4.3 Data Mode

#### 15.4.3.1 Raw RGB Mode

Two raw RGB modes, 16 bpp and 24 bpp, are used for transferring or receiving data. The data stream does not require more transformations, but does require reordering different panel resolutions. The mapping of the raw RGB signals and the panel types are listed in Table 15-1.

**Table 15-1. Signal Mapping for Raw RGB Modes and Panel Types**

| | 16 bpp | | | 24 bpp | | |
|---|---|---|---|---|---|---|
| | R | G | B | R | G | B |
| 24-bit panel | D15-11,000 | D10-5,00 | D4-0,000 | D23-16 | D15-8 | D7-0 |
| 18-bit panel | D15-11,0 | D10-5 | D4-0,0 | D23-18 | D15-10 | D7-2 |

For the 16 bpp raw RGB mode, the raw data in the memory formats of RGB444, RGB555, and RGB565 are listed in Table 15-2. If the format is set to RGB444, the LCD controller will automatically capture the R data from D[11:8] in the raw memory and translate to D'[15:11] with a postfix of 1'b0, the G data from D[7:4] in the raw memory and translate to D'[10:5] with a postfix of 2'b0, and the B data from D[3:0] in the raw memory and translate to D'[4:0] with a postfix of 1'b0. If the format is set to RGB555, D[14:10] will be captured and translated to D'[15:11] and the G data will be translated to D[10:5] with a postfix of 1'b0.

**Table 15-2. RGB444, RGB555, and RGB565 Memory Formats in 16 bpp Raw RGB Mode**

| Memory Format | 16 bpp | | |
|---|---|---|---|
| | R | G | B |
| RGB444 | D[11:8] | D[7:4] | D[3:0] |
| RGB555 | D[14:10] | D[9:5] | D[4:0] |
| RGB565 | D[15:11] | D[10:5] | D[4:0] |

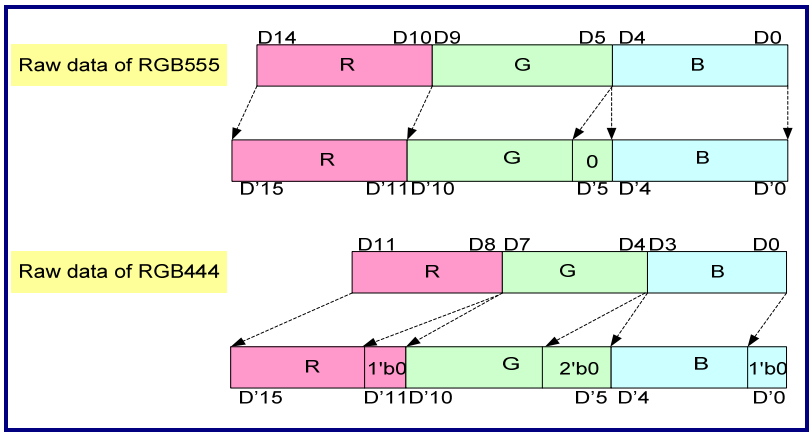



**Figure 15-5. Raw Data Translation for RGB444 and RGB555 Memory Formats**

### 15.4.3.2 YCbCr422 Mode

When the YCbCr422 mode is enabled, the value of 16 bpp cannot be changed. The pixel sequence is listed in Table 15-3.

**Table 15-3. Pixel Sequence in YCbCr422 Mode**

| | Y | Cb | Cr |
|---|---|---|---|
| Pixel 2N | $D_{(2N)15-8}$ | $D_{(2N)7-0}$ | - |
| Pixel 2N+1 | $D_{(2N+1)15-8}$ | - | $D_{(2N+1)7-0}$ |

### 15.4.3.3 YCbCr420 Mode

In the YCbCr420 mode, a bpp value of 8 cannot be changed. The memory locations for storing components, Y, Cb, and Cr, are separate. The base address of three frame buffers can be separately programmed in the register files. When the YCbCr420 data are read out from each individual FIFO, it is necessary to recover the chrominance values at each missed line. A line buffer is used to vertically store and interpolate the previous data and to estimate the absent chrominance value. A converted YcbCr422 data stream is then outputted to the next stage.

**Table 15-4. Component Locations/Sequences in YCbCr420 Mode**

| Component | Base Address of Frame Buffer | Sequence |
|---|---|---|
| Y | Image0_Base | $Y_{00}, Y_{01}, Y_{02}, Y_{03}, Y_{04}, Y_{05}$,..... |
| | | $Y_{10}, Y_{11}, Y_{12}, Y_{13}, Y_{14}, Y_{15}$,..... |
| | | $Y_{20}, Y_{21}, Y_{22}, Y_{23}, Y_{24}, Y_{25}$,..... |
| | | ..... |
| Cb | Image1_Base | $Cb_{00}, Cb_{02}, Cb_{04}, Cb_{06}, Cb_{08}$,..... |
| | | $Cb_{20}, Cb_{22}, Cb_{24}, Cb_{26}, Cb_{28}$,..... |
| | | ..... |
| Cr | Image2_Base | $Cr_{00}, Cr_{02}, Cr_{04}, Cr_{06}, Cr_{08}$,..... |
| | | $Cr_{20}, Cr_{22}, Cr_{24}, Cr_{26}, Cr_{28}$,..... |
| | | ..... |

### 15.4.3.4 RGB Palette Mode

The controller provides a remapping mode to enhance the flexibility of applications. Figure 15-6 depicts the basic conception of this concern. This scheme is able to perform a trade-off between the color variety and the memory space bandwidth requirement. Four memory space bandwidths, 8 bpp, 4 bpp, 2 bpp, and 1 bpp, are available to accommodate this trade-off process. A palette RAM is employed to store the look-up table and to regenerate the most desirable RGB components. The palette RAM holds up to 256 x 16-bit color values, which is physically structured as 128 x 32 bits. Table 15-5 lists the palette RAM entries. The pixel data from the input FIFO are used to address an palette location. The 1-bit pixel data addresses first two entries, the 2-bit pixel data addresses first four entries, the 4-bit pixel data addresses first 16 entries, and the 8-bit pixel data select any of the 256 palette entries. In the 16-bit or 24-bit pixel mode, the palette RAM will not be used and can be removed to reduce the hardware cost

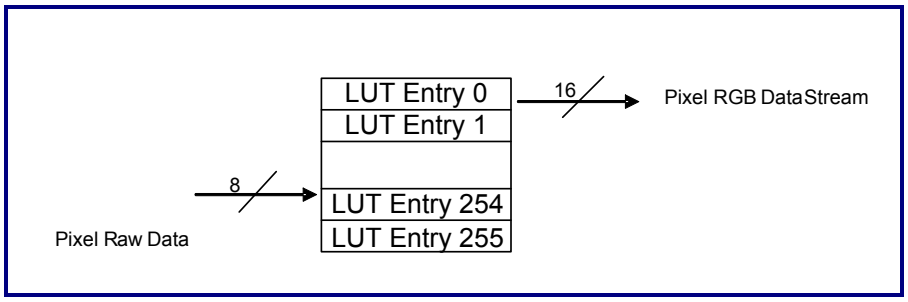



**Figure 15-6. Palette Remapping Operation**

**Table 15-5. Palette RAM Data Structure**

| Bit | Content | Description |
|---|---|---|
| [31:27] | R[4:0] | Red palette data for index 1 |
| [26:21] | G[5:0] | Green palette data for index 1 |
| [20:16] | B[4:0] | Blue palette data for index 1 |
| [15:11] | R[4:0] | Red palette data for index 0 |
| [10:5] | G[5:0] | Green palette data for index 0 |
| [4:0] | B[4:0] | Blue palette data for index 0 |

### 15.4.4 On-screen Display (Simple OSD)

#### 15.4.4.1 Font-based Architecture

The OSD window is made up of fonts. These fonts are generated from the database which is preloaded into the dedicate RAM. Each font database entry contains a 12x16 matrix describing the appearance of the font dot. Because these databases are stored in RAM, it can be modified easily to meet different requirements, such as icons and multi-languages applications. The RAM size for storing the font database is 4Kx12; it can store up to 256 fonts.

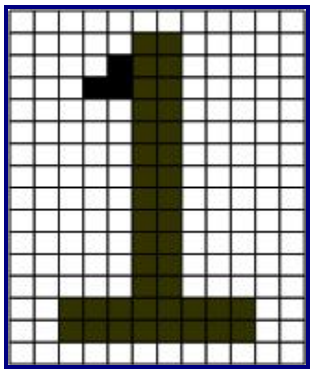


**Figure 15-7. Font Structure (12 x 16 Dots Per Font)**

### 15.4.4.2 Attribute Description

Each font displayed in the OSD window owns its unique attribute description. It is a 12-bit wide entry and can be programmed based on customer's preference. The font attribute structure is shown in Table 15-6. The value, $\mathbf{I_{[7:0]}}$, specifies which font is selected among the total 256 fonts to display on the window. The value, $\mathbf{F_{[1:0]}}$, addresses the foreground color from the 4-entry, 8-bit color palette. The value, $\mathbf{B_{[1-0]}}$, is used to handle the background color that is similar to the previous one. When **B** equals to zero, it indicates that a transparence operation is enabled, the color mapping is no longer necessary, and the font background makes the original image be partially visible on the screen. Only three palette colors entries are utilized for background control. The transparency level can be 25%, 50%, 75%, or 100% for the original image.

**Table 15-6. Font Attribute Structure**

| Font Index | | | | | | | | Foreground Color | | Background Color | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $I_7$ | $I_6$ | $I_5$ | $I_4$ | $I_3$ | $I_2$ | $I_1$ | $I_0$ | $F_1$ | $F_0$ | $B_1$ | $B_0$ |

### 15.4.4.3 OSD Window Control

The OSD engine provides functions to enhance the application flexibility. The window position of a screen is fully programmable in both the horizontal and vertical directions. An up-scaling function is implemented to control the font size independently by duplicating the font dot horizontally and vertically, based on the values of the control register. The scaling-up factor can be 1, 2, 3, or 4. The dimensions are also programmable for users to adjust the window aspect as required.

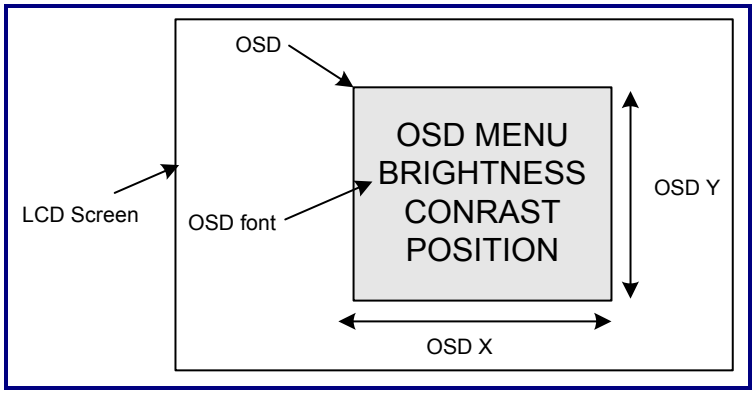



**Figure 15-8. OSD Basic Operation Description**

### 15.4.5 Scaler

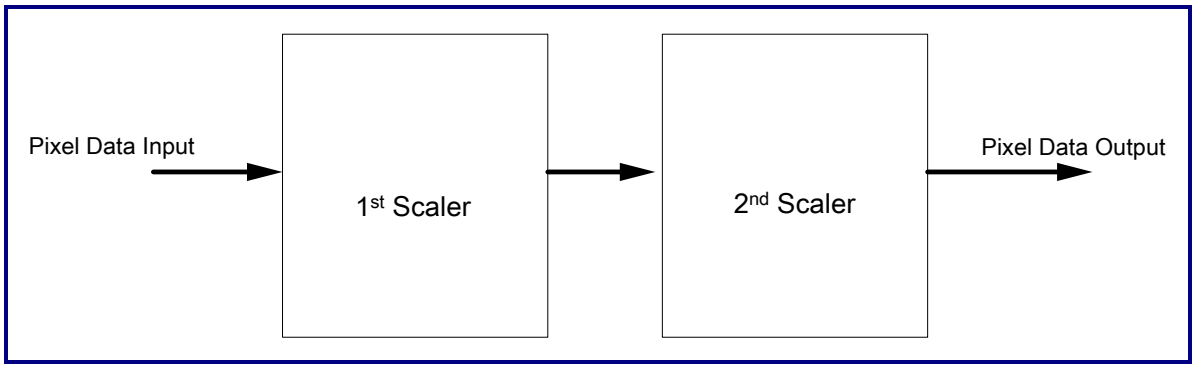



**Figure 15-9. Block Diagram of Scaler**

#### 15.4.5.1 First Scaler

The first scaler performs the down-scaling by the linear interpolation processing. The scaling down ratio is selectable among 1/2x1/2, 1/4x1/4, 1/8x1/8, 1/16x1/16, 1/32x1/32, 1/64x1/64, and 1/128x1/128. For example, the figure below is 1/2x1/2 down-scaling.

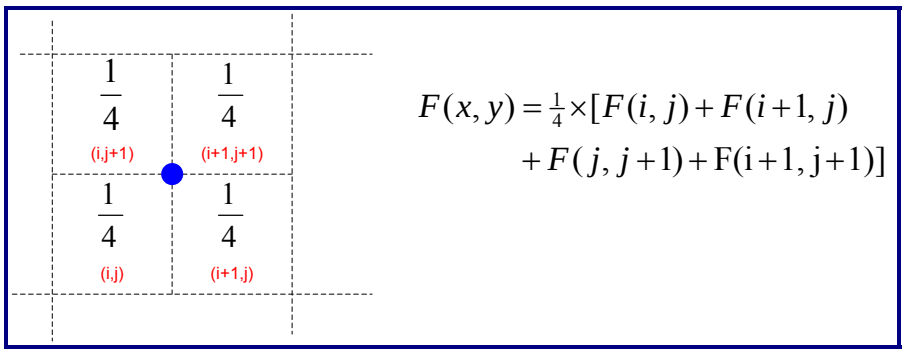



**Figure 15-10. 1/2x1/2 Down-scaling**

#### 15.4.5.2 Second Scaler

The second scaler performs the arbitrary ratio scaling. The scaling ratio can be any value between 1/2x1/2 and 2x2. The ratio is dependent on the input and output resolutions set by users (0x1100 ~ 0x110C). The scaling function provides three modes by the second scaler, which include the nearly bilinear mode, threshold nearly bilinear mode, and most neighborhood mode. The threshold nearly bilinear mode and most neighborhood mode can work only when down-scaling is applied.

#### 15.4.5.2.1 Nearly Bilinear Interpolation Mode

The output pixel is the interpolation from four input pixels. As shown in Figure 15-11, the result pixel (x,y) is located between the original image pixel(i, j+1), (i+1, j+1), (i, j), and (i+1, j). In this mode, the result pixel (x,y) is generated by the formulas shown in Figure 15-12.

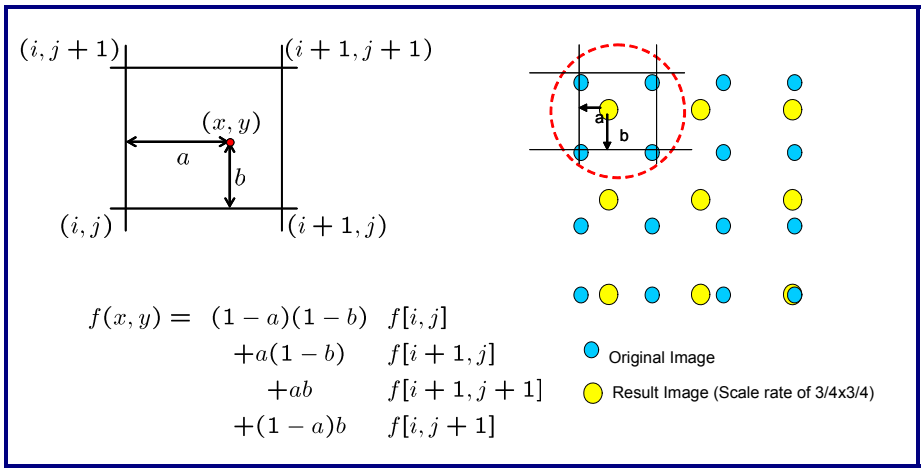



**Figure 15-11. Nearly Bilinear Interpolation**

#### 15.4.5.2.2 Threshold Nearly Bilinear Interpolation Mode

The threshold nearly bilinear interpolation mode uses the nearly bilinear interpolation with four threshold values: Hor_high_th, Hor_low_th, Ver_high_th, and Ver_low_th (Offset from 0x1114 to 0x1120). These four threshold values are used to devide the original image into nine locations. If the resulted pixel is located in the middle area of the original image, the resulted pixel will be produced by the nearly bilinear interpolation. Otherwise, the result pixel will come from the nearly original pixel value.

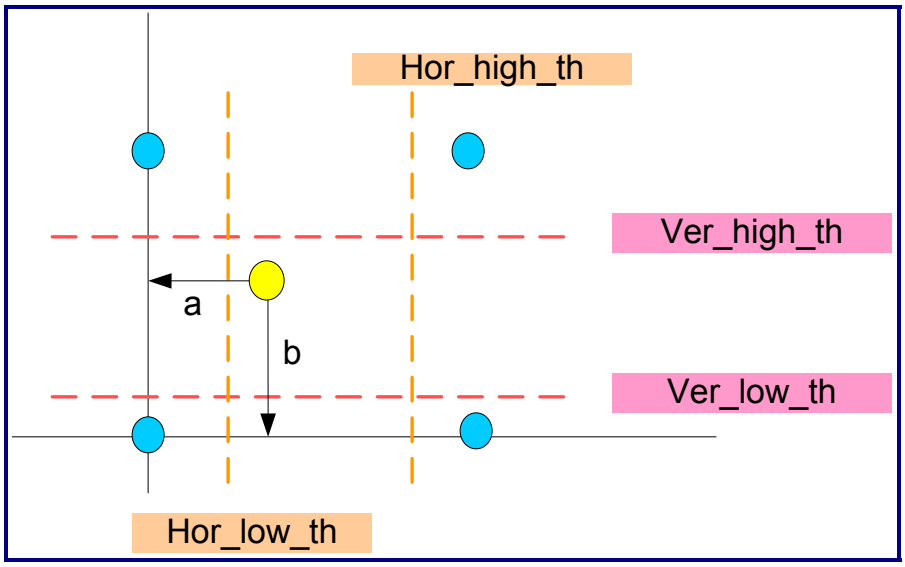



**Figure 15-12. Threshold Nearly Bilinear Interpolation**

#### 15.4.5.2.3 Most Neighborhood Mode

In this mode, the original image is divided into four average areas. The result pixeled value comes from the nearest original pixel value.

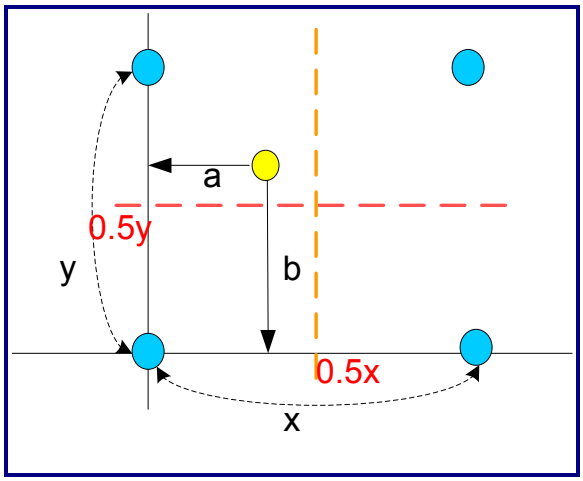



**Figure 15-13. Threshold Nearly Bilinear Interpolation**

### 15.4.6 Interrupt Controller

The interrupt controller first receives four internal interrupt signals, the AHB master error interrupt, FIFO under-run interrupt, vertical sync. interrupt, and frame base address updated interrupt. These signals are then combined to generate a global interrupt signal. These four interrupt signals can be enabled or disabled by programming the enable bit of the interrupt enable register. The combined interrupt will be asserted if any of these four interrupts is enabled and occurs.

## 15.5 Programming Model

### 15.5.1 Summary of Control Registers

**Table 15-7. Summary of Control Registers**

| **LCD Global Parameter** | | | | | |
|---|---|---|---|---|---|
| **Address** | **Type** | **Width** | **Reset Value** | **Name** | **Description** |
| 0x000 | R/W | 17 | 0x00 | Lcd_function_en | LCD function enable |
| 0x004 | R/W | 20 | 0x00 | Lcd_panel_pixel | LCD panel pixel parameter |
| 0x008 | R/W | 4 | 0x00 | Lcd_intr_mask | LCD interrupt enable mask |
| 0x00C | W | 4 | 0x00 | Lcd_intr_clr | LCD interrupt status clear |
| 0x010 | R | 4 | 0x00 | Lcd_intr_status | LCD interrupt status |
| 0x014 | R/W | 16 | 0x00 | Lcd_frame_buffer | LCD frame buffer parameter |
| 0x018 | R/W | 30 | 0x00 | Lcd_im0frm0base | LCD panel image0 frame0 base address |
| 0x024 | R/W | 30 | 0x00 | Lcd_im1frm0base | LCD panel image1 frame0 base address |
| 0x030 | R/W | 30 | 0x00 | Lcd_im2frm0base | LCD panel image2 frame0 base address |
| 0x03C | R/W | 30 | 0x00 | Lcd_im3frm0base | LCD panel image3 frame0 base address |
| 0x048 | R/W | 8 | 0x00 | Lcd_patbarcolor | LCD pattern generator |
| 0x04C | R/W | 32 | 0x04040404 | Lcd_fifoth | LCD FIFO threshold |
| 0x050 | R,W | 16 | 0x00 | GPIOControl | GPI/GPO control |
| **LCD timing and polarity parameters** | | | | | |
| 0x100 | R/W | 32 | 0x00 | Lcd_hortiming | LCD horizontal timing control |
| 0x104 | R/W | 26 | 0x00 | Lcd_vertiming0 | LCD vertical timing control0 |
| 0x108 | R/W | 8 | 0x00 | Lcd_vertiming1 | LCD vertical timing control1 |
| 0x10C | R/W | 12 | 0x00 | Lcd_polarity | LCD polarity control |
| **LCD output format parameters** | | | | | |
| 0x204 | R/W | 4 | 0x00 | Lcd_tv_parm0 | LCD TV parameters |
| 0x208 | R/W | 24 | 0x00 | Lcd_tv_parm1 | LCD TV line/cycle parameters |

| LCD Global Parameter | | | | | |
|---|---|---|---|---|---|
| **Address** | **Type** | **Width** | **Reset Value** | **Name** | **Description** |
| 0x20C | R/W | 24 | 0x00 | Lcd_tv_parm2 | LCD TV field polarity parameters |
| 0x210 | R/W | 20 | 0x00 | Lcd_tv_parm3 | LCD TV vertical blank0 ~ 1 |
| 0x214 | R/W | 20 | 0x00 | Lcd_tv_parm4 | LCD TV vertical blank2 ~ 3 |
| 0x218 | R/W | 24 | 0x00 | Lcd_tv_parm5 | LCD TV vertical active parameters |
| 0x21C | R/W | 20 | 0x00 | Lcd_tv_parm6 | LCD TV horizontal blank0 ~ 1 |
| 0x220 | R/W | 10 | 0x00 | Lcd_tv_parm7 | LCD TV horizontal blank2 |
| 0x224 | R/W | 12 | 0x00 | Lcd_tv_parm8 | LCD TV horizontal active parameters |
| 0x228 | R/W | 20 | 0x00 | Lcd_tv_parm9 | LCD TV vertical blank4 ~ 5 |
| 0x22C | R/W | 20 | 0x00 | Lcd_tv_parm10 | LCD TV VBI parameters |
| **LCD palette accessing port** | | | | | |
| 0xA00 ~ 0xBFC | W | 32 | - | PaletteWritePort | LCD palette RAM write accessing port |
| **Scaler control registers** | | | | | |
| 0x1100 | R/W | 12 | 0x00 | Scal_hor_no_in | Scaler horizontal resolution of input |
| 0x1104 | R/W | 12 | 0x00 | Scal_ver_no_in | Scaler vertical resolution of input |
| 0x1108 | R/W | 14 | 0x00 | Scal_hor_no_out | Scaler horizontal resolution of output |
| 0x110C | R/W | 14 | 0x00 | Scal_ver_no_out | Scaler vertical resolution of output |
| 0x1110 | R/W | 9 | 0x00 | Scal_misc | Scaler miscellaneous control registers |
| 0x1114 | R/W | 9 | 0x00 | Scal_hor_high_th | Horizontal up-scaling high threshold |
| 0x1118 | R/W | 9 | 0x00 | Scal_hor_low_th | Horizontal up-scaling low threshold |
| 0x111C | R/W | 9 | 0x00 | Scal_ver_high_th | Vertical up-scaling high threshold |
| 0x1120 | R/W | 9 | 0x00 | Scal_ver_low_th | Vertical up-scaling low threshold |
| 0x112C | R/W | 16 | 0x00 | Scal_hor_ver_num | Scaler resolution parameter |
| **Virtual registers** | | | | | |
| 0x1500 | R/W | 18 | 0x00 | vs_control | Virtual screen control |
| **Simple OSD control registers** | | | | | |
| 0x2000 | R/W | 15 | 0x00 | Osd_ctrl0 | OSD control register 0 |
| 0x2004 | R/W | 23 | 0x00 | Osd_ctrl1 | OSD control register 1 |
| 0x2008 | R/W | 32 | 0x00 | Osd_forecolor | OSD fore color register |
| 0x200C | R/W | 26 | 0x00 | Osd_backcolor | OSD back color register |
| 0x8000 ~ 0xBFFC | R/W | 12 | - | OSD Font accessing port | OSD font RAM accessing port |

## 15.5.2 LCD Global Parameters

### 15.5.2.1 LCD Function Enable Parameters (Offset: 0x0000)

**Table 15-8. LCD Function Enable Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| 16 | AddrSyn_En | R/W | Address sync. enable<br>When this bit is set to '1', the initial address of each DMA buffer can be updated to a new address by setting the "AddrUpdate" bit to '1.' When users want to update the base address of all DMA buffers at same time, this bit must be set to update the sync., address.<br>0: Disable the address synchronous<br>1: Enable the address synchronous |
| 15 | VBI_En | R/W | VBI enable<br>This bit informs the LCD controller to insert the VBI information from the DMA buffer3 to the vertical blanking lines. This bit is effective when TVEn is set to '1'.<br>0: Disable the VBI<br>1: Enable the VBI |
| 14 | PatGen | R/W | Test pattern generator<br>When the pattern generator is enabled, the LCD controller will automatically send out patterns to panel depending on the programmed value of 0x48.<br>0: Turn off the pattern generator<br>1: Turn on the pattern generator |
| 13 | TVEn | R/W | TV output function.<br>When this bit is set, the LCD controller will support the TV output.<br>0: Disable the TV output<br>1: Enable the TV output |
| 12 | - | - | Reserved |
| [11:10] | - | - | Reserved |
| [9:8] | - | - | Reserved |
| 7 | - | - | Reserved |
| 6 | - | - | Reserved |
| 5 | ScalerEn | R/W | Scaler enable<br>When this bit is set, the LCD controller will support the scaling function. The detailed scaling dimension depends on the scaler control registers.<br>0: Disable the scaling function<br>1: Enable the scaling function |

| Bit | Name | Type | Description |
|---|---|---|---|
| 4 | OSDEn | R/W | On-Screen Display (OSD) enable<br>When this bit is set, the LCD controller can support the OSD function. The displaying type of OSD is depending on programming the OSD control register.<br>0: Disable the On-Screen Display<br>1: Enable the On-Screen Display |
| 3 | EnYCbCr | R/W | YCbCr input mode control<br>When this bit is set, it informs the LCD controller that the data type in the frame buffer is the YCbCr format.<br>0: Enable the RGB format<br>1: Enable the YCbCr format |
| 2 | EnYCbCr420 | R/W | YCbCr420 input mode control<br>When this bit is set, it informs the LCD controller that the data type in the frame buffer is the YCbCr420 format.<br>0: Enable the YCbCr422 input mode<br>1: Enable the YCbCr420 input mode<br>This mode only functions when EnYCbCr is set to '1'. |
| 1 | LCDon | R/W | LCD screen on/off control<br>0: Disable the LCD screen (All data output pins are forced to 0.)<br>1: Enable the LCD screen (Normal operation) |
| 0 | LCDen | R/W | LCD controller enable control<br>0: Disable the LCD controller and force all LCD signals including synchronization and data to zero.<br>1: Enable the LCD controller (Normal operation) |

### 15.5.2.2 LCD Panel Pixel Parameters (Offset 0x0004)

**Table 15-9. LCD Panel Pixel Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| 19 | PRST | R/W | PCLK domain reset<br>The signal can reset all status in the APB clock domain. This register can only be cleared by users. |
| 18 | LRST | R/W | LC_CLK domain reset<br>The signal can reset all status in the LCD clock domain. This register can only be cleared by users. |
| 17 | MRST | R/W | Master domain reset AHB<br>The signal can reset all status in the AHB domain. This register can only be cleared by users. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 16 | AddrUpdate | R/W | Address update<br>This bit is valid when AddrSyn_En (0x0000) is set.<br>When AddrUpdate is set and AddrSyn_En = 1, the base address of each DMA buffer can be updated. |
| [15:14] | UpdateSrc | R/W | Update source selection<br>This signal selects the image source which the interrupt, “IntNxtBase”, is according to.<br>00: Image0<br>01: Image1<br>10: Image2<br>11: Image3 |
| [13:12] | - | - | Reserved |
| 11 | - | - | Reserved |
| [10:9] | Vcomp | R/W | Generate interrupt<br>00: Start of the vertical sync.<br>01: Start of the vertical back porch<br>10: Start of the vertical active image<br>11: Start of the vertical front porch |
| [8:7] | RGBTYPE | R/W | RGB input format<br>This bit identifies the input RGB format when BppFifo = 100.<br>00: RGB 565 input<br>01: RGB 555 input<br>10: RGB 444 input<br>11: Reserved |
| [6:5] | Endian | RW | Frame buffer data endianness control<br>00: Little-endian byte little-endian pixel<br>01: Big-endian byte big-endian pixel<br>10: Little-endian byte big-endian pixel (WinCE)<br>11: Reserved |
| 4 | BGRSW | R/W | RGB or BGR output format selection<br>0: RGB normal output<br>1: BGR red and blue swapped |
| 3 | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [2:0] | BppFifo | R/W | Pixel format in FIFO<br>This signal informs the data format in each DMA buffer (Bits per pixel).<br>000: 1 bpp<br>001: 2 bpp<br>010: 4 bpp<br>011: 8 bpp<br>100: 16 bpp<br>101: 24 bpp<br>Others: Reserved |

### 15.5.2.3 LCD Interrupt Enable Mask Parameters (Offset 0x0008)

**Table 15-10. LCD Interrupt Enable Mask Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| 3 | IntBusErrEn | R/W | AHB master error interrupt enable<br>This bit is set to enable the bus error interrupt.<br>1: Enable<br>0: Disable |
| 2 | IntVstatusEn | R/W | Vertical duration comparison interrupt enable<br>This bit is set to enable the vertical status interrupt.<br>1: Enable<br>0: Disable |
| 1 | IntNxtBaseEn | R/W | Next frame base address updated interrupt enable<br>This bit is set to enable the update base address successfully interrupt.<br>1: Enable<br>0: Disable |
| 0 | IntFIFOUdnEn | R/W | FIFO under-run interrupt enable<br>This bit is set to enable the FIFO under-run interrupt.<br>1: Enable<br>0: Disable |

### 15.5.2.4 LCD Interrupt Status Clear Parameters (Offset 0x000C)

**Table 15-11. LCD Interrupt Status Clear Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| 3 | ClrBusErr | W | Setting this bit to '1' clears the AHB master error interrupt status.<br>Setting this bit to '0' has no effect. |
| 2 | ClrVstatus | W | Setting this bit to '1' clears the interrupt status of vertical duration comparison.<br>Setting this bit to '0' has no effect. |
| 1 | ClrNxtBase | W | Setting this bit to '1' clears the frame buffer base address update interrupt status.<br>Setting this bit to '0' has no effect. |
| 0 | ClrFIFOUdn | W | Setting this bit to '1' clears the FIFO under-run interrupt status.<br>Setting this bit to '0' has no effect. |

### 15.5.2.5 LCD Interrupt Status Parameters (Offset 0x0010)

**Table 15-12. LCD Interrupt Status Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| 3 | IntBusErr | R | AHB master error status<br>This bit is set when the AMBA AHB master encounters a bus error response from a slave. |
| 2 | IntVstatus | R | Vertical comparison<br>When this bit is set, one of the four vertical durations is reached. This duration can be selected via the LCD control registers |
| 1 | IntNxtBase | R | Frame buffer base address update<br>When this bit is set, the current base address registers are uccessfully updated by the next address registers. |
| 0 | IntFIFOUdn | R | FIFO under-run<br>When this bit is set, the DMA FIFO is a read access for an under-run condition caused by empty. |

### 15.5.2.6 Frame Buffer Parameters (Offset 0x0014)

**Table 15-13. Frame Buffer Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| [15:14] | Im3ScalDown | R/W | Scaling down for image 3<br>The register is valid when POP is enabled. The image from "LCDImage0FrameBase" can be scaled down depending on the value.<br>00: Disable<br>01: The image 0 will scale down to 1/2 x 1/2.<br>10: The image 0 will scale down to 1/2 x 1.<br>11: Reserved |
| [13:12] | Im2ScalDown | R/W | Scaling down for image 2<br>The register is valid when POP enable. The image from "LCDImage2FrameBase" can be scaling down depended on the value.<br>00: Disable<br>01: The image 0 will scale down to 1/2 x 1/2.<br>10: The image 0 will scale down to 1/2 x 1.<br>11: Reserved |
| [11:10] | Im1ScalDown | R/W | Scaling down for image 1<br>The register is valid when POP enable. The image from "LCDImage1FrameBase" can be scaled down depending on the value.<br>00: Disable<br>01: The image 0 will scale down to 1/2 x 1/2.<br>10: The image 0 will scale down to 1/2 x 1.<br>11: Reserved |
| [9:8] | Im0ScalDown | R/W | Scaling down for image 0<br>The register is valid when POP enable or TV enable. The image from "LCDImage0FrameBase" can be scaling down depended on the value.<br>00: Disable<br>01: The image 0 will scale down to 1/2 x 1/2.<br>10: The image 0 will scale down to 1/2 x 1.<br>11: Reserved |
| [7:0] | Reserved | - | - |

### 15.5.2.7 LCD Panel Image0 Frame0 Base Address (Offset 0x0018)

**Table 15-14. Image0 Frame Base Address**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:2] | LCDImage0FrameBase | R/W | LCD frame0 base address of Image0<br>This is the start address of the frame data in memory. The two LSB bits are defaulted to 0. The total occupied memory address range is:<br>Starting address: {LCDImageFrame0Base, 0, 0} |

### 15.5.2.8 LCD Panel Image1 Frame0 Base Address (Offset 0x0024)

**Table 15-15. Image1 Frame Base Address**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:2] | LCDImage1FrameBase | R/W | LCD frame0 base address of Image1<br>This is the start address of the frame data in memory. The two LSB bits are defaulted to 0. The total occupied memory address range is:<br>Starting address: {LCDImage1Frame0Base, 0, 0} |

### 15.5.2.9 LCD Panel Image2 Frame0 Base Address (Offset 0x0030)

**Table 15-16. Image1 Frame Base Address**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | LCDImage2FrameBase | R/W | LCD frame0 base address of Image2<br>This is the start address of the frame data in memory. The two LSB bits are default 0. The total occupied memory address range is:<br>Starting address: {LCDImage2Frame0Base, 0, 0} |

### 15.5.2.10 LCD Panel Image3 Frame0 Base Address (Offset 0x003C)

**Table 15-17. Image0 Frame Base Address**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | LCDImage3FrameBase | R/W | LCD frame0 base address of Image3<br>This is the start address of the frame data in memory. The defaults of two LSB bits are 0s. The total occupied memory address range is:<br>Starting address: {LCDImage3Frame0Base, 0, 0}<br>※This base address is used for VBI when the TV is turned on and when the VBI is enabled. |

### 15.5.2.11 PatGen Pattern Bar Distance Parameter (Offset 0x0048)

**Table 15-18. PatGen Pattern Bar Distance Parameter**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:6] | Img3PatGen | R/W | Pattern generator of Image 3<br>The LCD controller will generate the different test patterns of image 3, depending on the value.<br>00: Vertical color bar<br>01: Horizontal color bar<br>10: Single color<br>11: Reserved |
| [5:4] | Img2PatGen | R/W | Pattern generator of Image 2<br>The LCD controller will generate the different test patterns of image 2, depending on the value.<br>00: Vertical color bar<br>01: Horizontal color bar<br>10: Single color<br>11: Reserved |
| [3:2] | Img1PatGen | R/W | Pattern generator of Image 1<br>The LCD controller will generate the different test patterns of image 1, depending on the value.<br>00: Vertical color bar<br>01: Horizontal color bar<br>10: Single color<br>11: Reserved |
| [1:0] | Img0PatGen | R/W | Pattern Generator of Image 0<br>The LCD controller will generate the different test patterns of image 0, depending on the value.<br>00: Vertical color bar<br>01: Horizontal color bar<br>10: Single color<br>11: Reserved |

### 15.5.2.12 FIFO Threshold Control Parameter (Offset 0x004C)

**Table 15-19. FIFO Threshold Control Parameter**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | Buf3Threshold | R/W | DMA buffer 3 threshold<br>The buffer 3 will assert request for master to fetch data in when the buffer3's empty count is larger than this value. This threshold must be larger or equal to 4. |
| [23:16] | Buf2Threshold | R/W | DMA buffer 2 threshold<br>The buffer 2 will assert request for master to fetch data in when the buffer2's empty count is larger than this value. This threshold must be larger or equal to 4. |
| [15:8] | Buf1Threshold | R/W | DMA buffer 1 threshold<br>The buffer 1 will assert request for master to fetch data in when the buffer0's empty count is larger than this value. This threshold must be larger or equal to 4. |
| [7:0] | Buf0Threshold | R/W | DMA buffer 0 threshold<br>The buffer 0 will assert request for master to fetch data in when the buffer0's empty count is larger than this value. This threshold must be larger or equal to 4. |

## 15.5.3 LCD Timing and Polarity Parameters

### 15.5.3.1 LCD Horizontal Timing Control Parameter (Offset 0x0100)

**Table 15-20. LCD Horizontal Timing Control Parameter**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | HBP | R/W | **Horizontal Back Porch**<br>**Actual horizontal back porch = HBP + 1**<br>HBP is the number of the LC_PCLK periods between the falling edge of LC_PCLK and the start of active data. The 8-bit HBP field is used to specify the number of pixel clock periods inserted at the beginning of each line or row of pixels. After the line clock for the previous line has been de-asserted, the value in HBP is used to count the number of pixel clocks to wait before starting the next display line. HBP can generate a delay of 1 to 256 pixel clock cycles. |
| [23:16] | HFP | R/W | **Horizontal Front Porch**<br>**Actual horizontal front porch = HFP + 1**<br>HFP is the number of the LC_PCLK periods between the end of active data and the rising edge of LC_HS. The 8-bit HFP field sets the number of pixel clock intervals at the end of each line or row of pixels, before the LCD line clock is pulsed. Once a complete line of pixels is transmitted to the LCD driver, the value in HFP is used to count the number of pixel clocks to wait before asserting the line clock. HFP can generate a period of 1 to 256 pixel clock cycles. |

| Bit | Name | Type | Description |
|---|---|---|---|
| [15:8] | HW | R/W | **Horizontal Synchronization Pulses Width**<br>**Actual horizontal synchronization pulses width = HW + 1**<br>HW is the width of the LC_HS signal in the LC_PCLK periods. The 8-bit HW field specifies the pulse width of the line clock in the passive mode, or the horizontal synchronization pulse in the active mode. |
| [7:0] | PL | R/W | **Pixels-per-Line**<br>**Actual pixels-per-line = 16 * (PL + 1)**<br>The PL bit field specifies the number of pixels in each line of row of the screen. PL is a 6-bit value that represents between 16 and 2048 PL. PL is used to count the number of pixel clock that occur before the HFP is applied (Program the value required divided by 16, minus 1). |

### 15.5.3.2 LCD Vertical Timing Control Parameters (Offset 0x0104)

**Table 15-21. LCD Vertical Timing Control Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | VFP | R/W | **Vertical Front Porch**<br>**Actual vertical front porch = VFP**<br>VFP is the number of inactive lines at the end of a frame prior to the vertical synchronization period. The 8-bit VFP field is used to specify the number of line clocks to insert at the end of each frame. Once a complete frame of pixels is transmitted to the LCD display, the value in VFP will be used to count the number of line clock periods to wait. After the count has elapsed the vertical synchronization (LC_VS) signal is asserted. VFP generates from 0 to 255-line clock cycles. |
| [23:22] | - | - | Reserved |
| [21:16] | VW | R/W | **Vertical Synchronization Pulse Width**<br>**Actual vertical synchronization pulse width = VW + 1**<br>VW is the number of horizontal synchronization lines. Program to the number of lines required minus one. The 6-bit VW field is used to specify the pulse width of the vertical synchronization pulse. The register is programmed with the number of line clocks in vertical synchronization period minus one. Number of horizontal synchronization lines. |
| [15:12] | - | - | Reserved |
| [11:0] | LF | R/W | **Lines-per-Frame**<br>**Actual line-per-frame = LF + 1**<br>LF is the number of active lines per frame. The LF field specifies the total number of lines or rows on the LCD panel being controlled. The value of LF is 11-bit allowing between 1 and 2048 lines. |

### 15.5.3.3 LCD Vertical Back Porch Parameter (Offset 0x0108)

**Table 15-22. LCD Vertical Back Porch Parameter**

| Bit | Name | Type | Description |
|---|---|---|---|
| [7:0] | VBP | R/W | **Vertical Back Porch**<br>**Actual vertical back porch = VBP**<br>VBP is the number of inactive lines at the start of a frame subsequent to the vertical synchronization period. The 8-bit VBP field is used to specify the number of line clocks inserted at the beginning of a frame. The VBP count starts just after the vertical synchronization signal of the previous frame being negated for active mode. After this has occurred, the count value in VBP sets the number of line clock periods inserted before the next frame. VBP generates from 1 to 256 extra line clock cycles. |

### 15.5.3.4 LCD Polarity Control Parameters (Offset 0x010C)

**Table 15-23. LCD Polarity Control Parameters**

| Bit | Name | Type | Description |
|---|---|---|---|
| [14:8] | DivNo | R/W | LCD panel clock divisor control (LC_PCLK)<br>LC_PCLK is divided from LC_CLK.<br>The actual divisor value is equal to (DivNo + 1). |
| [7:5] | - | - | Reserved |
| 4 | - | - | Reserved |
| 3 | IDE | R/W | 0: LC_DE/LC_AC output pin is active high.<br>1: LC_DE/LC_AC output pin is active low.<br>The invert output enable bit is used to select the active polarity of the output enable signal. |
| 2 | ICK | R/W | 0: Data is driven on the LCD data lines on the rising edge of LC_PCLK.<br>1: Data is driven on the LCD data lines on the falling edge of LC_PCLK.<br>The ICK bit is used to select the edge of the panel clock on which pixel data is driven out onto the LCD data lines. |
| 1 | IHS | R/W | 0: LC_HS/LC_LP pin is active high and inactive low.<br>1: LC_HS/LC_LP pin is active low and inactive high.<br>The invert horizontal sync bit is used to select the active polarity of the horizontal sync signal. |
| 0 | IVS | R/W | 0: LC_VS/LC_FLM pin is active high and inactive low.<br>1: LC_VS/LC_FLM pin is active low and inactive high.<br>The invert vertical sync bit is used to select the active polarity of the vertical sync signal. |

### 15.5.4 LCD Output Format Parameters

#### 15.5.4.1 LCD TV Parameters (Offset 0x0204)

This parameter is valid when TVEn is set.

Notes:

1. When TVEn is set, the scaler function must be disabled for the interlace input or the interlace output format.
2. The total line number, LF (Offset 0x104) and horizontal pixel number, PL (Offset 0x100) must be set.

**Table 15-24. LCD TV Parameter0**

| Bit | Name | Type | Description |
|---|---|---|---|
| 3 | BusWidth | R/W | TV data width selection<br>0: Output data is 8-bit wide.<br>1: Output data is 16-bit wide. |
| 2 | ImgFormat | R/W | Interlace/Progress Image format selection<br>The value decides the image in the frame buffer is the format of “Progress” or “Interlace”.<br>0: Input image format is progress.<br>1: Input image format is interlace. |
| 1 | PHASE | R/W | TVE clock phase<br>0: The TV clock is the same as LC_PCLK.<br>1: The TV clock is the inversion of LC_PCLK. |
| 0 | OutFormat | R/W | Interlace/Progress output format selection<br>0: Interlace output<br>1: Progressive output (When the output format is progressive, Imgformat must be set progress) |

#### 15.5.4.2 TV Control Parameters (Offset 0x0208)

Figure 15-14 and Figure 15-15 are the vertical and horizontal timing diagrams of the TV interface. Users can program the parameters based on the timing diagrams. Users can program any timing parameter for BT656 and BT1120. As shown in Figure 15-14, the active field includes three regions. The “V_ACT0/1” region which is the real image, and the “V_BLK3” regions. This region represents that the height of the raw images does not meet the TV vertical resolution. The LCD controller can patch black line above/below the image to meet TV’s vertical resolution. Figure 15-15 shows that the active line includes three regions, the

"H_ACT0" region which is the real image, and the "H_BLK1" and "H_BLK2" regions. This represents if the width of raw image does not meet the TV horizontal resolution, the LCD controller can patch black pixel left/right the image to meet the TV horizontal resolution.

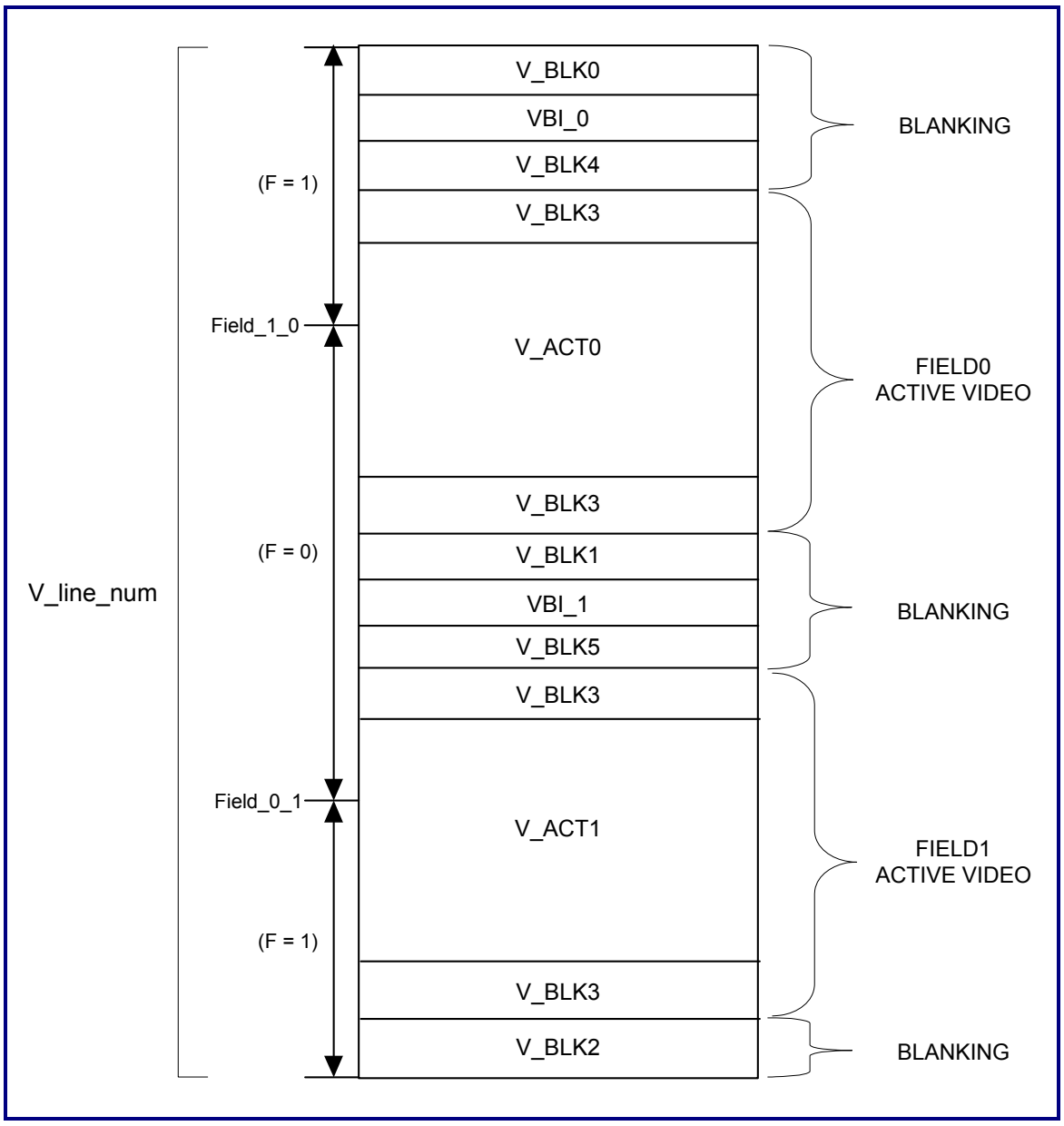



**Figure 15-14. Vertical Timing of BT656/BT1120**

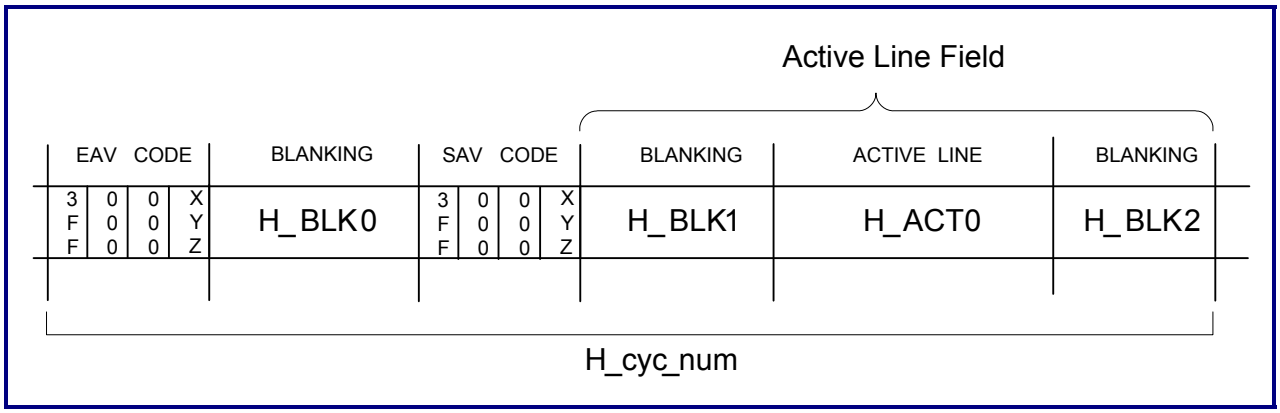



**Figure 15-15. Horizontal Timing of BT656/BT1120**

**Table 15-25. TV Parameter 1**

| Bit | Name | Type | Description |
|---|---|---|---|
| [23:12] | V_line_num | R/W | Vertical line number<br>Take NTSC for example, the value of this parameter must be set to 525, and it must be set to 625 for the PAL system. |
| [11:0] | H_cyc_num | R/W | Horizontal cycle number.<br>Take NTSC for example, the value of this parameter must be set to 1716, and it must be set to 1728 for the PAL system. |

### 15.5.4.3 TV Field Polarity Parameters (Offset 0x020C)

**Table 15-26. TV Parameter 2**

| Bit | Name | Type | Description |
|---|---|---|---|
| [23:12] | Field_0_1 | R/W | The field indicates which line the value of “F” changes from 0 to 1. |
| [11:0] | Field_1_0 | R/W | The field indicates which line the value of “F” changes from 1 to 0. |

### 15.5.4.4 TV Vertical Blank Parameters (Offset 0x0210)

**Table 15-27. TV Parameter 3**

| Bit | Name | Type | Description |
|---|---|---|---|
| [21:12] | V_Blk1 | R/W | This field indicates the number of line of the V_BLK1 region. |
| [11:10] | - | - | Reserved |
| [9:0] | V_Blk0 | R/W | This field indicates the number of line of the V_BLK0 region. |

### 15.5.4.5 TV Vertical Blank Parameters (Offset 0x0214)

**Table 15-28. TV Parameter 4**

| Bit | Name | Type | Description |
|---|---|---|---|
| [21:12] | V_Blk3 | R/W | This field indicates the number of line of the V_BLK3 region. |
| [11:10] | - | - | Reserved |
| [9:0] | V_Blk2 | R/W | This field indicates the number of line of the V_BLK2 region. |

### 15.5.4.6 TV Vertical Active Parameters (Offset 0x0218)

**Table 15-29. TV Parameter 5**

| Bit | Name | Type | Description |
|---|---|---|---|
| [23:12] | V_Act1 | R/W | This parameter indicates the active number of line of field 1. |
| [11:0] | V_Act0 | R/W | This parameter indicates the active number of line of field 0. |

### 15.5.4.7 TV Horizontal Blank Parameters (Offset 0x021C)

**Table 15-30. TV Parameter 6**

| Bit | Name | Type | Description |
|---|---|---|---|
| [21:12] | H_Blk1 | R/W | This field shows the number of cycle of the H_BLK1 region. |
| [11:10] | - | - | Reserved |
| [9:0] | H_Blk0 | R/W | This field shows the number of cycle of the H_BLK0 region. |

### 15.5.4.8 TV Horizontal Blank Parameters (Offset 0x0220)

**Table 15-31. TV Parameter 7**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:10] | - | - | Reserved |
| [9:0] | H_Blk2 | R/W | This field shows the number of cycle of the H_BLK2 region. |

### 15.5.4.9 TV Horizontal Active Parameters (Offset 0x0224)

**Table 15-32. TV Parameter 8**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:12] | - | - | Reserved |
| [11:0] | H_Act0 | R/W | This field shows the active number of cycle of each line. |

### 15.5.4.10 TV Vertical Blank Parameters (Offset 0x0228)

**Table 15-33. TV Parameter 9**

| Bit | Name | Type | Description |
|---|---|---|---|
| [21:12] | V_Blk5 | R/W | This field shows the number of line of the V_BLK5 region. |
| [11:10] | - | - | Reserved |
| [9:0] | V_Blk4 | R/W | This field shows the number of line of the V_BLK4 region. |

### 15.5.4.11 TV VBI Parameters (Offset 0x022C)

The VBI data is fetched from the frame buffer 3. The LCDImage3FrameBase register must be set to the start address of the VBI data.

Users are suggested to set the VBI frame buffer base address at the register, 03CH. The VBI data format is used with the YCbCr422 format for data packing. If the TV image input format is progressive, the VBI data must be progressive in the frame buffer. If the TV input format is interlaced, the LCD controller will treat the VBI data format as interlace.

**Table 15-34. TV Parameter 10**

| Bit | Name | Type | Description |
|---|---|---|---|
| [21:12] | VBI_1 | R/W | This field shows the line number of the VBI_1 region. |
| [11:10] | - | - | Reserved |
| [9:0] | VBI_0 | R/W | This field shows the line number of the VBI_0 region.<br>VBI_0 must be the same as VBI_1. |

### 15.5.5 Scaler Control Registers

The scaler can not work when TV is enable and the input and output format is not progressive.
If users turn on TV and if the input format or output format is interlace, please use the 14H register to perform simple scaling.

#### 15.5.5.1 Horizontal Resolution Register of Scaler Input (Offset 0x1100)

**Table 15-35. Horizontal Input Resolution (Offset 0x1100)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:12] | - | - | Reserved |
| [11:0] | Hor_no_in | R/W | **Horizontal Resolution of Input Image**<br>Actual horizontal resolution of input image = Hor_no_in + 1<br>This field identifies the input image horizontal resolution and must be programmed to a non-zero value before the scaler becomes active. The value is between 1 and 2047. |

#### 15.5.5.2 Vertical Resolution Register of Scaler Input (Offset 0x1104)

**Table 15-36. Vertical Input Resolution (Offset 0x1104)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:12] | - | - | Reserved |
| [11:0] | Ver_no_in | R/W | **Vertical Resolution of Input Image**<br>Actual vertical resolution of input image = Ver_no_in + 1<br>This field identifies the input image vertical resolution and must be programmed as a non-zero value before the scaler becomes active. This field is programmed to the required number of ver_no_in minus 1. The value is between 1 and 2047. |

#### 15.5.5.3 Horizontal Resolution Register of Scaler Output (Offset 0x1108)

**Table 15-37. Horizontal Output Resolution (Offset 0x1108)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:14] | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [13:0] | Hor_no_out | R/W | **Horizontal Resolution of Output Image**<br>Actual horizontal resolution of output image = Hor_no_out<br>This field identifies the output image horizontal resolution and must be programmed as a non-zero value before the scaler becomes active. The value is between 1 and 4096. |

### 15.5.5.4 Vertical Resolution Register of Scaler Output (Offset 0x110C)

**Table 15-38. Vertical Output Resolution (Offset 0x110C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:14] | - | - | Reserved |
| [13:0] | Ver_no_out | R/W | **Vertical Resolution of Output Image**<br>Actual vertical resolution of output image = Ver_no_out<br>This field identifies the output image vertical resolution and must be programmed as a non-zero value before the scaler becomes active. The value is between 1 and 4096. |

### 15.5.5.5 Miscellaneous Control Register (Offset 0x1110)

**Table 15-39. Scaler Control Register (Offset 0x1110)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:14] | - | - | Reserved |
| [8:6] | Fir_sel | R/W | The scaling ratio selection of the 1st stage scaler<br>000: Bypass 1st stage scaler<br>001: Scaling down to 1/2x1/2<br>010: Scaling down to 1/4x1/4<br>011: Scaling down to 1/8x1/8<br>100: Scaling down to 1/16x1/16<br>101: Scaling down to 1/32x1/32<br>110: Scaling down to 1/64x1/64<br>111: Scaling down to 1/128x1/128 |
| 5 | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [4:3] | Hor_inter_mode | R/W | The horizontal interpolation mode<br>00: Nearly bilinear mode<br>The following modes are only available on down-scaling:<br>01: Threshold nearly bilinear mode<br>10: Most neighborhood mode<br>11: Reserved |
| [2:1] | Ver_inter_mode | R/W | The vertical interpolation mode<br>00: Nearly bilinear mode<br>The following modes are only available on down-scaling:<br>01: Threshold nearly bilinear mode<br>10: Most neighborhood mode<br>11: Reserved |
| 0 | Bypass_mode | R/W | This bit identifies the 2nd stage scaler bypass mode and bypasses the scaler. This bit must be set in the initial state and should not be changed once it is set. This bit must be cleared to activate the scaler. |

### 15.5.5.6 Horizontal High Threshold Register (Offset 0x1114)

**Table 15-40. Horizontal High Threshold Register (Offset 0x1114)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:9] | - | - | Reserved |
| [8:0] | Hor_high_th | R/W | Horizontal up-scaling high threshold register<br>The up-scaling coefficient is highly limited by this threshold value in the horizontal direction. The real value equals to Hor_high_th/256. |

### 15.5.5.7 Horizontal Low Threshold Register (Offset 0x1118)

**Table 15-41. Horizontal Low Threshold Register (Offset 0x1118)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:9] | - | - | Reserved |
| [8:0] | Hor_low_th | R/W | Horizontal up-scaling low threshold register<br>The up-scaling coefficient is low limited by this threshold value in the horizontal direction. The real value equals to Hor_low_th/256. |

### 15.5.5.8 Vertical High Threshold Register (Offset 0x111C)

**Table 15-42. Vertical High Threshold Register (Offset 0x111C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:9] | - | - | Reserved |
| [8:0] | Ver_high_th | R/W | Vertical up-scaling high threshold register<br>The up-scaling coefficient is high limited by this threshold value in the vertical direction. The real value equals to Ver_high_th/256. |

### 15.5.5.9 Vertical Low Threshold Register (Offset 0x1120)

**Table 15-43. Vertical Low Threshold Register (Offset 0x1120)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:9] | - | - | Reserved |
| [8:0] | Ver_low_th | R/W | Vertical up-scaling low threshold register<br>The up-scaling coefficient is low limited by this threshold value in the vertical direction. The real value equals to Ver_low_th/256. |

### 15.5.5.10 Scaler Resolution Parameters (Offset 0x112C)

The parameters, scal_hor_num and scal_ver_num, are calculated as shown below, where ver_no_in/hor_no_in represents either the input image resolution register value if the first scaler is bypassed or the output image resolution of the first scaler minus one if the first scaler is involved.

**Scaling up:**

Scal_ver_num = [(ver_no_in+1)/ver_no_out]x256

Scal_hor_num = [(hor_no_in+1)/hor_no_out]x256

*If the scaling up ratio is 2x2, and if Scal_ver_num and Scal_hor_num are ignored.*

**Scaling down:**

Scal_ver_num = [mod((ver_no_in+1)/ver_no_out)]x256/ver_no_out

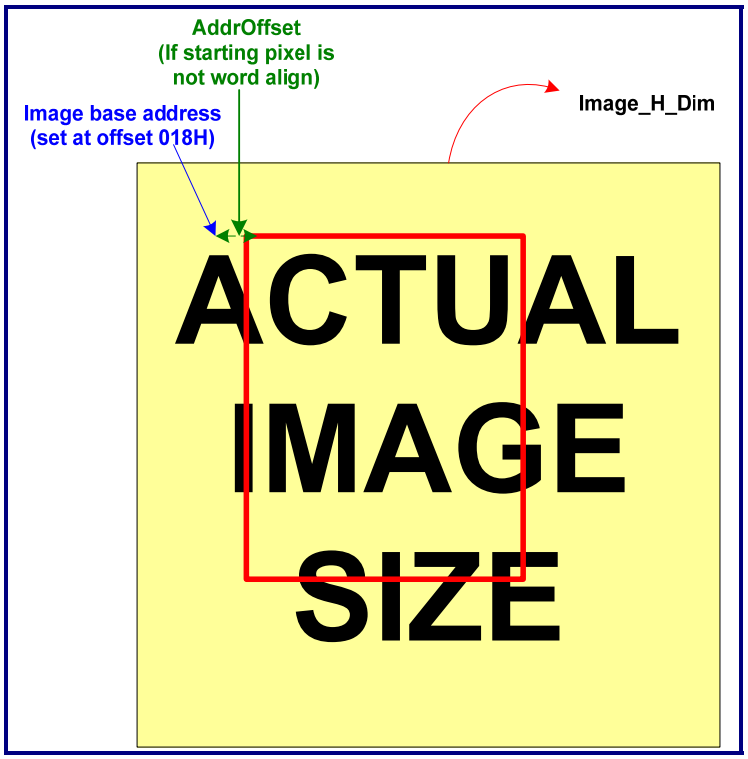
Scal_hor_num = [mod((hor_no_in+1)/hor_no_out)]x256/hor_no_out

**Table 15-44. Scaler Resolution Parameters (Offset 0x112C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | Scal_hor_num | R/W | The second-stage initial numerator of the scaler coefficient in the horizontal direction. |
| [7:0] | Scal_ver_num | R/W | The second-stage initial numerator of the scaler coefficient in the vertical direction. |

### 15.5.6 Virtual Screen

The LCD controller provides the hardware horizontal/vertical scrolling function. The image base address (0x018) and address offset register (0x1500) must be updated if the image is scrolled. The horizontal/vertical width of the scrolled image is set in PL (0x0100)/LF (0x0104). The image stored in the frame buffer should be larger than the screen size of the LCD panel.



### 15.5.6.1 Virtual Screen Control Register (Offset 0x1500)

**Table 15-45. Virtual Screen Control Register (Offset 0x1500)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| 20 | VirtualScreenEn | R/W | Virtual screen enable<br>0: Disable<br>1: Enable |
| [19:17] | - | - | Reserved |
| [16:12] | AddrOffset | R/W | In the 32-bit AMBA data bus, the starting address of a screen equals to “LCDImage0FrameBase” + “AddrOffset”. The value of AddrOffset must be multiple of “bppfifo”. |
| [11:0] | Image_H_Dim | R/W | The horizontal dimension of an image stored in the horizontal frame buffer (Pixels per line) |

## 15.5.7 OSD Control Registers for Simple OSD

### 15.5.7.1 OSD Scaling and Dimension Control (Offset 0x2000)

**Table 15-46. OSD Scaling and Dimension Control (Offset 0x2000)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:18] | - | - | Reserved |
| [17:12] | OSDHdim | R/W | This field specifies the horizontal dimension of the OSD window, that is, the total font number of each row |
| [11:9] | - | - | Reserved |
| [8:4] | OSDVdim | R/W | This field specifies the vertical dimension of the OSD window, that is, the total vertical font number |
| [3:2] | OSDHScal | R/W | Define the horizontal up-scaling factor<br>This field duplicates the dot several times, based on the set values below:<br>00: Original horizontal size<br>01: Enlarge the window two times, horizontally<br>10: Enlarge the window three times, horizontally<br>11: Enlarge the window four times, horizontally |

| Bit | Name | Type | Description |
|---|---|---|---|
| [1:0] | OSDVScal | R/W | Define the vertical up-scaling factor<br>This field duplicates a line several times based on the value set below:<br>00: Original vertical size<br>01: Enlarge the vertical size of a window two times<br>10: Enlarge the vertical size of a window three times<br>11: Enlarge the vertical size of a window four times |

### 15.5.7.2 OSD Position Control (Offset 0x2004)

**Table 15-47. OSD Position Control (Offset 0x2004)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | - | - | Reserved |
| [23:12] | OSDHPos | R/W | This field specifies the horizontal position of the OSD window. The reference original position is the left boundary of the LCD screen. The valid value is ranging from 1 to 1280. However, a fixed offset value of 3 between OSDHPos and the actual window position is recommended. |
| 11 | - | - | Reserved |
| [10:0] | OSDVPos | R/W | This field specifies the vertical position of the OSD window. The reference original position is the upper boundary of the LCD screen. The valid value is ranging from 1 to 1280. |

### 15.5.7.3 OSD Foreground Color Control (Offset 0x2008)

**Table 15-48. OSD Foreground Color Control (Offset 0x2008)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | OSDFrPal3 | R/W | OSD foreground color palette entry 3<br>The format can be set below:<br>{$D_{31-30}$, 6{& $D_{31-30}$}} forms MSB of the Y component.<br>{$D_{29-27}$, 5{$D_{29-27}$}} forms MSB of the Cb component.<br>{$D_{26-24}$, 5{$D_{26-24}$}} forms MSB of the Cr component. |
| [23:16] | OSDFrPal2 | R/W | OSD foreground color palette entry 2<br>The operation is similar to entry 3. |
| [15:8] | OSDFrPal1 | R/W | OSD foreground color palette entry 1<br>The operation is similar to entry 3. |
| [7:0] | OSDFrPal0 | R/W | OSD foreground color palette entry 0<br>The operation is similar to entry 3. |

### 15.5.7.4 OSD Background Color Control (Offset 0x200C)

**Table 15-49. OSD Background Color Control (Offset 0x200C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | OSDBgPal3 | R/W | OSD foreground color palette entry 3.<br>The format can be set below:<br>$\{D_{31-30}, 6\{\& D_{31-30}\}\}$ forms MSB of the Y component.<br>$\{D_{29-27}, 5\{D_{29-27}\}\}$ forms MSB of the Cb component.<br>$\{D_{26-24}, 5\{D_{26-24}\}\}$ forms MSB of the Cr component. |
| [23:16] | OSDBgPal2 | R/W | OSD foreground color palette entry 2<br>The operation is similar to entry 3. |
| [15:8] | OSDBgPal1 | R/W | OSD foreground color palette entry 1<br>The operation is similar to entry 3. |
| [7:2] | - | - | Reserved |
| [1:0] | OSDTrans | R/W | OSD background transparency control<br>00: 25% transparency<br>01: 50% transparency<br>10: 75% transparency<br>11: 100% transparency |

### 15.5.8 OSD Font Database Write Accessing Port (Offset 0x8000 ~ 0xBFFC)

Each font database is stored as amatrix of 12x16 in the font-RAM. The RAM is organized as 12-bit in width. It requires 16 entries (192 bits) to record the font content. A bit that equals 1 refers to the foreground, and a bit that equals 0 refers to background. The total size of a font memory is 4Kx12, so the total number of fonts that can be loaded is 256. There is an example for a font '1' at the location 0x8000 ~ 0x803C

**Table 15-50. OSD Font Database Write Accessing Port (Offset 0x8000 ~ 0xBFFC)**

| Address Offset | Bit | Type | Description | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $D_{11}$ | $D_{10}$ | $D_9$ | $D_8$ | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
| 0x8000 (0) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0x8004 (1) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8008 (2) | 11:0 | W | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x800C (3) | 11:0 | W | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8010 (4) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8014 (5) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8018 (6) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x801C (7) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8020 (8) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8024 (9) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8028 (10) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x802C (11) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8030 (12) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0x8034 (13) | 11:0 | W | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0x8038 (14) | 11:0 | W | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0x803C (15) | 11:0 | W | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 15.5.9 OSD Window Attribute Write Accessing Port (Offset 0xC000 ~ 0xC7FC)

Fonts are used to build an OSD window. The font dimensions are defined by programming the associated registers. Each font in the window owns a unique attribute to describe its behavior. The format is a 12-bit wide entry, as mentioned earlier. These attributes are saved in an attribute RAM and are sequentially placed, starting from the up-left corner, so that it can move from left to right and row by row. The offset address of the first attribute entry is 0xC000, and the last is 0xC7FC. Therefore, it is possible to display an OSD window with up to 512 fonts, regardless of the window aspect. Since the total font variety is 256, some fonts will be reused to build such a big window.

**Table 15-51. OSD Window Attribute Write Accessing Port (Offset 0xC000 ~ 0xC7FC)**

| Address | Bit | Type | Description | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $D_{11}$ | $D_{10}$ | $D_9$ | $D_8$ | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
| 0xC000 | 11:0 | W | Font index of the first font | | | | | | | | Foreground palette index of the first font | | Background palette index of the first font | |
| …..... | …. | …. | ..... | | | | | | | | ….. | | ….. | |
| 0xC7FC | 11:0 | W | Font index of the last font | | | | | | | | Foreground palette index of the last font | | Background palette index of the last font | |

# Chapter 16

# 10/100 Ethernet Controller

This chapter contains the following sections:


- 16.1 General Description
- 16.2 Features
- 16.3 Interface Clocking
- 16.4 Register Definition
- 16.5 Multicast Address Hash Table 1 Register, MAHT1, 32'h0 (Offset: 0x14 ~ 0x17)
- 16.6 Function Description
- 16.7 Initialization/Application Information
- 16.8 MII Management Interface
- 16.9 Multicast Address Hash Table Filtering

## 16.1 General Description

TMAC110 is a high-quality 10/100 Ethernet controller with DMA function. It includes an AHB wrapper, DMA engine, on-chip memories (TX FIFO and RX FIFO), MAC, and MII or RMII interface.

FTMAC110 is an Ethernet controller that provides AHB master capability and is fully compliant with the IEEE 802.3 100 Mbps and 10 Mbps specifications. The FTMAC110 DMA controller handles all data transfers between system memory and on-chip memories. The DMA engine supports the zero-copy data transfer that drastically improves the system performance. The DMA engine can be used to reduce the CPU loading, maximize the performance, and minimize the FIFO size. FTMAC110 has on-chip memories for buffering, which requires no external local buffer memory. The MII interface or RMII interface can support two specific data rates, 10 Mbps and 100 Mbps. The functionality is identical at both data rates, and so is the signal timing relationship. The only difference between the 10 Mbps and 100 Mbps operations is the nominal clock frequency.

## 16.2 Features

- Compliant with AMBA AHB revision 2.0: Supports AHB master and slave bus interface
- Supports pin to determine the endianness of AHB bus interface
- Supports DMA engine for transmitting and receiving packets
- Supports zero-copy data transfer
- Supports programmable AHB burst size
- Supports transmit and receive interrupt mitigation
- Supports Wake-On-LAN function and three wake-up events: Link status change, Magic packet and Wake-up frame
- Supports four Wake-On-LAN signals (Active high, active low, positive pulse, and negative pulse)
- Contains independent TX and RX FIFOs (2 kB each)
- Supports half and full duplex modes
- Supports flow control for full duplex and back pressure for half duplex
- Selects MII or RMII (Reduced MII) interface

## 16.3 Interface Clocking

FTMAC110 requires the following clock input sources:

- X_MII_TXCK: The transmit MII clock with frequencies up to 2.5/25 MHz (10 Mbps/100 Mbps mode)
- X_MII_RXCK: The receive MII clock with frequencies up to 2.5/25 MHz (10 Mbps/100 Mbps mode), or the RMII reference clock with frequencies up to 50 MHz.
- MAC_CLK: The MII clock with frequencies up to 25 MHz (100 Mbps mode), the clock is from PLL1_out/(MACCLK_PVALUE + 1).
- HCLK: The AHB bus clock

## 16.4 Register Definition

The following abbreviations are used for register definitions:

R/W: Read/Write
RC: Read Clear
RO: Read Only
R/W1C: Read/Write 1 Clear

### 16.4.1 Interrupt Status Register, ISR, 32'h0 (Offset: 0x00 ~ 0x03)

**Table 16-1. Interrupt Status Register, ISR, 32'h0 (Offset: 0x00 ~ 0x03)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| 9 | RC | 1'h0 | PHYSTS_CHG | PHY link status change |
| 8 | RC | 1'h0 | AHB_ERR | AHB error |
| 7 | RC | 1'h0 | RPKT_LOST | Received packet lost due to RX FIFO full |
| 6 | RC | 1'h0 | RPKT_SAV | Packets received into RX FIFO successfully |
| 5 | RC | 1'h0 | XPKT_LOST | Packets transmitted to Ethernet lost due to late collision or excessive collision |
| 4 | RC | 1'h0 | XPKT_OK | Packets transmitted to Ethernet successfully |
| 3 | RC | 1'h0 | NOTXBUF | Transmit buffer unavailable |
| 2 | RC | 1'h0 | XPKT_FINISH | TXDMA has moved data into the TX FIFO |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| 1 | RC | 1'h0 | NORXBUF | Receive buffer unavailable |
| 0 | RC | 1'h0 | RPKT_FINISH | RXDMA has received packets to RX buffer successfully |

### 16.4.2 Interrupt Enable Register, IME, 32'h0 (Offset: 0x04 ~ 0x07)

**Table 16-2. Interrupt Enable Register, IME, 32'h0 (Offset: 0x04 ~ 0x07)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| 9 | R/W | 1'h0 | PHYSTS_CHG_EN | Interrupt enable of ISR[9] |
| 8 | R/W | 1'h0 | AHB_ERR_EN | Interrupt enable of ISR[8] |
| 7 | R/W | 1'h0 | RPKT_LOST_EN | Interrupt enable of ISR[7] |
| 6 | R/W | 1'h0 | RPKT_SAV_EN | Interrupt enable of ISR[6] |
| 5 | R/W | 1'h0 | XPKT_LOST_EN | Interrupt enable of ISR[5] |
| 4 | R/W | 1'h0 | XPKT_OK_EN | Interrupt enable of ISR[4] |
| 3 | R/W | 1'h0 | NOTXBUF_EN | Interrupt enable of ISR[3] |
| 2 | R/W | 1'h0 | XPKT_FINISH_EN | Interrupt enable of ISR[2] |
| 1 | R/W | 1'h0 | NORXBUF_EN | Interrupt enable of ISR[1] |
| 0 | R/W | 1'h0 | RPKT_FINISH_EN | Interrupt enable of ISR[0] |

### 16.4.3 MAC Most Significant Address Register, MAC_MADR, 32'h0 (Offset: 0x08 ~ 0x0B)

**Table 16-3. MAC Most Significant Address Register, MAC_MADR, 32'h0 (Offset: 0x08 ~ 0x0B)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| [15:0] | R/W | 16'h0 | MAC_MADR | The most significant two bytes of MAC address |

### 16.4.4 MAC Least Significant Address Register, MAC_LADR, 32'h0 (Offset: 0x0C ~ 0x0F)

**Table 16-4. MAC Least Significant Address Register, MAC_LADR, 32'h0 (Offset: 0x0C ~ 0x0F)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | MAC_LADR | The least significant four bytes of MAC address |

### 16.4.5 Multicast Address Hash Table 0 Register, MAHT0, 32'h0 (Offset: 0x10 ~ 0x13)

**Table 16-5. Multicast Address Hash Table 0 Register, MAHT0, 32'h0 (Offset: 0x10 ~ 0x13)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | MAHT0 | Multicast address hash table bytes[3:0] (Hash table [31:0]) |

## 16.5 Multicast Address Hash Table 1 Register, MAHT1, 32'h0 (Offset: 0x14 ~ 0x17)

**Table 16-6. Multicast Address Hash Table 1 Register, MAHT1, 32'h0 (Offset: 0x14 ~ 0x17)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | MAHT1 | Multicast address hash table bytes[3:0] (Hash table [63:32]) |

### 16.5.1 Transmit Poll Demand Register, TXPD, 32'h0 (Offset: 0x18 ~ 0x1B)

**Table 16-7. Transmit Poll Demand Register, TXPD, 32'h0 (Offset: 0x18 ~ 0x1B)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | WO | 32'h0 | TXPD | When writing any value to the register, FTMAC110 will read the transmit descriptor to process and check the txdma_own bit, if txdma_own = '1'. FTMAC110 will then move the transmit buffer data into TX FIFO. The read value of the register is always 0. |

### 16.5.2 Receive Poll Demand Register, RXPD, 32'h0 (Offset: 0x1C ~ 0x1F)

**Table 16-8. Receive Poll Demand Register, RXPD, 32'h0 (Offset: 0x1C ~ 0x1F)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | WO | 32'h0 | RXPD | When writing any value to the register, FTMAC110 will read the receive descriptor to process and check the rxdma_own bit, if rxdma_own = '1'. FTMAC110 will then move the receive packet data from RX FIFO into the receive buffer in the system memory. The read value of the register is always 0. |

### 16.5.3 Transmit Ring Base Address Register, TXR_BADR, 32'h0 (Offset: 0x20 ~ 0x23)

**Table 16-9. Transmit Ring Base Address Register, TXR_BADR, 32'h0 (Offset: 0x20 ~ 0x23)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | TXR_BADR | Base address of the transmit ring<br>The base address must be 16-byte alignment. FTMAC110 will treat the base address bits[3:0] as 0 when reading descriptors if bits[3:0] are not zero. |

### 16.5.4 Receive Ring Base Address Register, RXR_BADR, 32'h0 (Offset: 0x24 ~ 0x27)

**Table 16-10. Receive Ring Base Address Register, RXR_BADR, 32'h0 (Offset: 0x24 ~ 0x27)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | RXR_BADR | Base address of the receive ring<br>The base address must be 16-byte alignment. FTMAC110 will treat the base address bits[3:0] as 0 when reading descriptors if bits[3:0] are not zero. |

### 16.5.5 Interrupt Timer Control Register, ITC, 32'h0 (Offset: 0x28 ~ 0x2B)

**Table 16-11. Interrupt Timer Control Register, ITC, 32'h0 (Offset: 0x28 ~ 0x2B)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| 15 | R/W | 1'h0 | TXINT_TIME_SEL | This field defines the period of TX cycle time.<br>When this bit is set, the TX cycle time is<br>100 Mbps mode → 81.92 µs<br>10 Mbps mode → 819.2 µs<br>When cleared, TX cycle time is<br>100 Mbps mode → 5.12 µs<br>10 Mbps mode → 51.2 µs |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [14:12] | R/W | 3'h0 | TXINT_THR | This field defines the maximum number of transmit interrupts that can be pending before an interrupt is generated.<br>When TXINT_THR is not equal to 0, FTMAC110 will issue a transmit interrupt when the transmit packet number transmitted by FTMAC110 reaches TXINT_THR.<br>When TXINT_THR = 0 and TXINT_CNT = 0, whether to issue a transmit interrupt or not will depend on TXIC in TXDES1. |
| [11:8] | R/W | 4'h0 | TXINT_CNT | This field defines the maximum wait time to issue the transmit interrupt after a packet has been transmitted by FTMAC110. The time unit is 1 TX cycle time.<br>When TXINT_CNT = 0, the function will be disabled. When TXINT_THR = 0 and TXINT_CNT = 0, whether to issue a transmit interrupt not will depend on TXIC in TXDES1. |
| 7 | R/W | 1'h0 | RXINT_TIME_SEL | This field defines the period of the RX cycle time.<br>When this bit is set, the RX cycle time will be:<br>100 Mbps mode → 81.92 µs<br>10 Mbps mode → 819.2 µs<br>When this bit is cleared, the RX cycle time will be<br>100 Mbps mode → 5.12 µs<br>10 Mbps mode → 51.2 µs |
| [6:4] | R/W | 3'h0 | RXINT_THR | This field defines the maximum number of receive interrupts that can be pending before an interrupt is generated. When RXINT_THR is not equal to '0', FTMAC110 will issue one receive interrupt when the receive packet number received by FTMAC110 reaches RXINT_THR.<br>If RXINT_THR = 0 and RXINT_CNT = 0, a receive interrupt will be issued when FTMAC110 finishes receiving a receive packet. |
| [3:0] | R/W | 4'h0 | RXINT_CNT | This field defines the maximum wait time to issue the receive interrupt after a packet has been received by FTMAC110. The time unit is 1 RX cycle time.<br>When RXINT_CNT = 0, the function will be disabled.<br>If RXINT_THR = 0 and RXINT_CNT = 0, a receive interrupt will be issued when a packet is received by FTMAC110. |

Recommended value = 32'h0000_1010

The Interrupt Timer Control Register allows the software driver to reduce the number of transmit interrupt (ISR[4]) and receive interrupt (ISR[0]) by setting the register. This can lower the CPU utilization for handling a large number of interrupts.

This register defines two threshold values for the receive packet number, transmit packet number, and two associated timers. The threshold value defines the maximum number of the receive interrupt or transmit interrupt that can be pending before an interrupt is generated. The timer defines the maximum wait time to issue transmit/receive interrupt after a packet has been transmitted/received by FTMAC110. The threshold value and timer combination allows for the batching of several packets into a single interrupt with a limit for how long it can be pending. This can prevent throughput from being impeded in heavy traffic, while the time limit prevents resources from being held for too long in low traffic.

The mitigation mechanism is similar for both the receive and transmit interrupts. There is a counter (TXPKT_CNT) inside FTMAC110 to count the packet transmitted by FTMAC110. When the counter reaches TXINT_THR and TXINT_THR is not equal to '0', FTMAC110 will issue a transmit interrupt. There is also a counter (RXPKT_CNT) in FTMAC110 to count the packet received by FTMAC110. When the counter reaches RXINT_THR and RXINT_THR is not equal to '0', FTMAC110 will issue receive interrupt. TXPKT_CNT is cleared when transmit interrupt is issued. RXPKT_CNT is cleared when receive interrupt is issued.

The following table lists the conditions for FTMAC110 to issue a transmit interrupt.

| TXINT_THR = 0 | TXINT_CNT = 0 | FTMAC110 Action |
|---|---|---|
| True | True | • Issues transmit interrupt after packet is transmitted and TXIC of the packet is set.<br>• Clears TXPKT_CNT |
| True | False | • Issues transmit interrupt after a packet is transmitted and timer reaches the value of TXINT_CNT<br>• Clears TXPKT_CNT |
| False | True | • Issues transmit interrupt if TXPKT_CNT = TXINT_THR<br>• Clears TXPKT_CNT |
| False | False | • Issues transmit interrupt if the following condition holds:<br>TXPKT_CNT = TXINT_THR<br>TXPKT_CNT = 1 and timer reaches the f TXINT_CNT value<br>• Clears TXPKT_CNT |

The following table lists the conditions for FTMAC110 to issue a transmit interrupt.

| RXINT_THR = 0 | RXINT_CNT = 0 | FTMAC110 Action |
|---|---|---|
| True | True | • Issues receive interrupt after packet is received by FTMAC110<br>• Clears RXPKT_CNT |
| True | False | • Issues receive interrupt after packet is received by FTMAC110 and timer reaches the value of RXINT_CNT<br>• Clears RXPKT_CNT |
| False | True | • Issues receive interrupt if RXPKT_CNT = RXINT_THR<br>• Clears TXPKT_CNT |
| False | False | • Issues receive interrupt if the following condition holds:<br>RXPKT_CNT = RXINT_THR<br>RXPKT_CNT = 1 and timer reaches the value of RXINT_CNT<br>• Clears RXPKT_CNT |

### 16.5.6 Automatic Polling Timer Control Register, APTC, 32'h0, (Offset: 0x2C ~ 0x2F)

The Automatic Polling Timer Control Register allows FTMAC110 to automatically poll the descriptors, which lower the CPU utilization. When transmit automatic poll function is enabled, FTMAC110 would automatically poll the transmit descriptor when transmit automatic poll timer expires. If the function is disabled, software needs to write Transmit Poll Demand Register (Offset: 0x18 ~ 0x1B) to trigger FTMAC110 to read the transmit descriptors after software prepared the transmit packets in transmit buffers.

When receive automatic poll function is enabled, FTMAC110 would automatically poll the receive descriptor when receive automatic poll timer expires. If the function is disabled, software needs to write Receive Poll Demand Register (Offset: 0x1C ~ 0x1F) to trigger FTMAC110 to read receive descriptors after software released the receive descriptors to FTMAC110.

**Table 16-12. Automatic Polling Timer Control Register, APTC, 32'h0, (Offset: 0x2C ~ 0x2F)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:13] | - | - | - | Reserved |
| 12 | R/W | 1'h0 | TXPOLL_TIME_SEL | This bit defines the period of TX poll time.<br>When this bit is set, the TX poll time will be<br>100 Mbps mode → 81.92 µs<br>10 Mbps mode → 819.2 µs<br>When this bit is cleared, the TX poll time will be<br>100 Mbps mode → 5.12 µs<br>10 Mbps mode → 51.2 µs |
| [11:8] | R/W | 4'h0 | TXPOLL_CNT | This field defines the period of transmit automatic polling time. The unit is 1 TX poll time. When TXPOLL_CNT is not equal to 0, FTMAC110 will poll the transmit descriptor automatically. If TXPOLL_CNT = 0, FTMAC110 will not poll the transmit descriptor automatically. |
| [7:5] | - | - | - | Reserved |
| 4 | R/W | 1'h0 | RXPOLL_TIME_SEL | This bit defines the period of the RX poll time.<br>When tis bit is set, the RX poll time will be<br>100 Mbps mode → 81.92 µs<br>10 Mbps mode → 819.2 µs<br>When this bit is cleared, the RX poll time will be<br>100 Mbps mode → 5.12 µs<br>10 Mbps mode → 51.2 µs |
| [3:0] | R/W | 4'h0 | RXPOLL_CNT | This field defines the period of receive automatic polling time. The unit is 1 RX poll time. When RXPOLL_CNT is not equal to 0, FTMAC110 would poll the receive descriptor automatically. If RXPOLL_CNT = 0, FTMAC110 would not poll the receive descriptor automatically. |

Recommended value = 32'h0000_0001

### 16.5.7 DMA Burst Length and Arbitration Control Register, DBLAC, 32'h0, (Offset: 0x30 ~ 0x33)

**Table 16-13. DMA Burst Length and Arbitration Control Register, DBLAC, 32'h0, (Offset: 0x30 ~ 0x33)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| [15:14] | R/W | 2'h0 | INCR_SEL | This field defines the maximum data byte count of a burst when FTMAC110 issues the INCR burst command in AHB bus.<br>2'b0x: No limitation<br>2'b10: The maximum data byte count is 32x4 bytes.<br>2'b11: The maximum data byte count is 64x4 bytes. |
| [13:10] | - | - | - | Reserved |
| 9 | R/W | 1'h0 | RX_THR_EN | Enable RX FIFO threshold arbitration |
| [8:6] | R/W | 3'h0 | RXFIFO_HTHR | The RX FIFO high threshold value for arbitration<br>When the used space in the RX FIFO is larger than or equal to the RX FIFO high threshold value, the RXDMA has higher priority than the TXDMA in using the DMA channel. The RXDMA keeps higher priority until the used space in the RX FIFO is less than or equal to the RX FIFO low threshold value. TXDMA then gets higher priority than the RXDMA. Software must be set. Users must set RXFIFO_HTHR larger than RXFIFO_LTHR to keep FTMAC110 work correctly. The unit is 256 bytes.<br>3'd0: Threshold = 0<br>3'd1: Threshold = 1/8 space of RX FIFO, 256 bytes<br>3'd2: Threshold = 2/8 space of RX FIFO, 512 bytes<br>3'd3: Threshold = 3/8 space of RX FIFO, 768 bytes<br>3'd4: Threshold = 4/8 space of RX FIFO, 1024 bytes<br>3'd5: Threshold = 5/8 space of RX FIFO, 1280 bytes<br>3'd6: Threshold = 6/8 space of RX FIFO, 1536 bytes<br>3'd7: Threshold = 7/8 space of RX FIFO, 1792 bytes |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [5:3] | R/W | 3'h0 | RXFIFO_LTHR | The RX FIFO low threshold value for arbitration<br>When the used space in the RX FIFO is less than or equal to the RX FIFO low threshold value, TXDMA has higher priority than the RXDMA when using the DMA channel. The unit is 256 bytes.<br>3'd0: Threshold = 0<br>3'd1: Threshold = 1/8 space of RX FIFO, 256 bytes<br>3'd2: Threshold = 2/8 space of RX FIFO, 512 bytes<br>3'd3: Threshold = 3/8 space of RX FIFO, 768 bytes<br>3'd4: Threshold = 4/8 space of RX FIFO, 1024 bytes<br>3'd5: Threshold = 5/8 space of RX FIFO, 1280 bytes<br>3'd6: Threshold = 6/8 space of RX FIFO, 1536 bytes<br>3'd7: Threshold = 7/8 space of RX FIFO, 1792 bytes |
| 2 | R/W | 1'h0 | INCR16_EN | This field defines if FTMAC110 could use INCR16 burst command in AHB bus. |
| 1 | R/W | 1'h0 | INCR8_EN | This field defines if FTMAC110 could use INCR8 burst command in AHB bus. |
| 0 | R/W | 1'h0 | INCR4_EN | This field defines if FTMAC110 could use INCR4 burst command in AHB bus. |

Recommended value = 32'h0000_0390

There are some limitations for use of INCR4, INCR8, and INCR16. Users must observe these limitations, as follows:

| AHB Bus Clock Range | Limitation on AHB Bus Side of SDRAM Controller |
|---|---|
| 25 (25 MHz not included) ~ 15 MHz | hready must assert low at least one AHB bus clock during every 3 data phases |
| 50 (50 MHz not included) ~ 25 MHz | hready must assert low at least one AHB bus clock during every 7 data phases |
| 75 (75 MHz not included) ~ 50 MHz | hready must assert low at least one AHB bus clock during every 15 data phases |
| 100 (100 MHz not included) ~ 75 MHz | hready must assert low at least one AHB bus clock during every 21 data phases |
| 125 (125 MHz not included) ~ 100 MHz | hready must assert low at least one AHB bus clock during every 31 data phases |
| 133 MHz ~ 125 MHz | hready must assert low at least one AHB bus clock during every 39 data phases |

Because FTMAC110 always uses WORD as the transfer size on the AHB bus, the limitation above on the data phases only applies to the WORD-size transfer. If the system fails to meet the limitation, INCR4/INCR8/INCR16 will not be enabled when FTMAC110 is used. In this case, FTMAC110 can only use INCR to transfer data in such a system. Users must ensure that the limitation above is met in the system before enabling INCR4/INCR8/INCR16 when using FTMAC110.

### 16.5.8 Revision Register, REVR, 32'h0, (Offset: 0x34 ~ 0x37)

**Table 16-14. Revision Register, REVR, 32'h0, (Offset: 0x34 ~ 0x37)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:16] | RO | - | REV_B1 | The first digit of reversion tag<br>Example: If the revision tag is version_1_2_r3, then the filed is 1. |
| [15:8] | RO | - | REV_B2 | The second digit of reversion tag<br>Example: If the revision tag is version_1_2_r3, then the filed is 2. |
| [7:0] | RO | - | REV_B3 | The third digit of reversion tag<br>Example: If the revision tag is version_1_2_r3, then the filed is 3. |

The register is only for internal use in Faraday.

### 16.5.9 Feature Register, FEAR, 32'h0, (Offset: 0x38 ~ 0x3B)

**Table 16-15. Feature Register, FEAR, 32'h0, (Offset: 0x38 ~ 0x3B)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:3] | - | - | - | Reserved |
| 2 | RO | - | HALFE | This bit supports the half duplex feature. |
| 1 | RO | - | FULLF | This bit supports the full duplex feature. |
| 0 | RO | - | WOLF | This bit supports the Wake-On-LAN feature. |

The register is only for internal use in Faraday.

### 16.5.10 Reserved Register, (Offset: 0x3C ~ 0x87)

### 16.5.11 MAC Control Register, MACCR, 32'h0, (Offset: 0x88 ~ 0x8B)

**Table 16-16. MAC Control Register, MACCR, 32'h0, (Offset: 0x88 ~ 0x8B)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:19] | - | - | - | Reserved |
| 18 | R/W | 1'h0 | SPEED_100 | Speed mode control<br>1: 100 Mbps<br>0: 10 Mbps |
| 17 | R/W | 1'h0 | RX_BROADPKT | Receives broadcast packet |
| 16 | R/W | 1'h0 | RX_MULTIPKT | Receives all multicast packets |
| 15 | R/W | 1'h0 | FULLDUP | Full duplex |
| 14 | R/W | 1'h0 | CRC_APD | This bit appends CRC to transmitted packet. |
| 13 | - | - | - | Reserved |
| 12 | R/W | 1'h0 | RCV_ALL | This bit does not check the destination address of the incoming packet. |
| 11 | R/W | 1'h0 | RX_FTL | This bit stores the incoming packet even if its length is great than 1518 bytes. |
| 10 | R/W | 1'h0 | RX_RUNT | This bit stores the incoming packet even if its length is less than 64 bytes. |
| 9 | R/W | 1'h0 | HT_MULTI_EN | This bit enables storing the incoming packet if the packet passes hash table address filtering and is a multicast packet. |
| 8 | R/W | 1'h0 | RCV_EN | Receiver enable |
| 7 | - | 1'h0 | - | Reserved |
| 6 | R/W | 1'h0 | ENRX_IN_HALFTX | This bit enables the packet reception when transmitting packet in the half-duplex mode. |
| 5 | R/W | 1'h0 | XMT_EN | Transmitter enable |
| 4 | R/W | 1'h0 | CRC_DIS | This bit disables the CRC check when receiving packets |
| 3 | R/W | 1'h0 | LOOP_EN | Internal loop-back |
| 2 | R/W | 1'h0 | SW_RST | Software reset<br>Writing 1 to this bit will enable software reset. Software reset will last 64 AHB bus clocks and then activate auto-clear. |
| 1 | R/W | 1'h0 | RDMA_EN | Enable receive DMA channel<br>If this bit is zero, reception will be stopped immediately. |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| 0 | R/W | 1'h0 | XDMA_EN | Enable transmit DMA channel<br>If this bit is zero, transmission will be stopped immediately. |

## 16.5.12 MAC Status Register, MACSR, 32'h0, (Offset: 0x8C ~ 0x8F)

**Table 16-17. MAC Status Register, MACSR, 32'h0, (Offset: 0x8C~ 0x8F)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| 11 | RC | 1'h0 | COL_EXCEED | Collision amount exceeds 16 |
| 10 | RC | 1'h0 | LATE_COL | Transmitter detects late collision |
| 9 | RC | 1'h0 | XPKT_LOST | Packets transmitted to Ethernet lost due to late collision or excessive collision. |
| 8 | RC | 1'h0 | XPKT_OK | Packets transmitted to Ethernet successfully |
| 7 | RC | 1'h0 | RUNT | Receiver detects a runt packet |
| 6 | RC | 1'h0 | FTL | Receiver detects a frame that is too long. |
| 5 | RC | 1'h0 | CRC_ERR | Incoming packet CRC check result is invalid, unless the CRC_DIS bit is set. |
| 4 | RC | 1'h0 | RPKT_LOST | Received packets lost due to RX FIFO full |
| 3 | RC | 1'h0 | RPKT_SAVE | Packets received to RX FIFO successfully |
| 2 | RC | 1'h0 | COL | Incoming packet dropped due to collision |
| 1 | RC | 1'h0 | BROADCAST | Incoming packet for broadcast address |
| 0 | RC | 1'h0 | MULTICAST | Incoming packet for multicast address |

## 16.5.13 PHY Control Register, PHYCR, 32'h0, (Offset: 0x90 ~ 0x93)

**Table 16-18. PHY Control Register, PHYCR, 32'h0, (Offset: 0x90 ~ 0x93)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| 27 | R/W | 1'h0 | MIIWR | This bit initializes a write sequence to PHY by setting this bit to 1. The bit will be automatically cleared after the write operation is finished. |
| 26 | R/W | 1'h0 | MIIRD | This bit initializes a read sequence to PHY by setting this bit to 1. The bit will be automatically cleared after the read operation is finished. |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [25:21] | R/W | 5'h0 | REGAD | PHY register address |
| [20:16] | R/W | 5'h0 | PHYAD | PHY address |
| [15:0] | RO | 16'h0 | MIIRDATA | Read data from PHY |

### 16.5.14 PHY Write Data Register, PHYWDATA, 32'h0, (Offset: 0x94 ~ 0x97)

**Table 16-19. PHY Write Data Register, PHYWDATA, 32'h0, (Offset: 0x94 ~ 0x97)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| [15:0] | R/W | 16'h0 | MIIWDATA | Writes data to PHY |

### 16.5.15 Flow Control Register, FCR, 32'h0000_A400, (Offset: 0x98 ~ 0x9B)

**Table 16-20. Flow Control Register, FCR, 32'h0000_A400, (Offset: 0x98 ~ 0x9B)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | R/W | 15'h0 | PAUSE_TIME | Pauses time in pause frame. The unit is 512 bit times. The 512 bit times is:<br>100 Mbps mode → 5.12 µs<br>10 Mbps mode → 51.2 µs |
| [15:12] | R/W | 4'hA | FC_HIGH | RX FIFO free space high threshold<br>Send a pause frame with pause time = 32'h0000 when RX FIFO free space is larger than the high threshold.<br>The unit is 128 bytes. Default value is 4'hA. |
| [11:8] | R/W | 4'h4 | FC_LOW | RX FIFO free space low threshold<br>Send a pause frame with pause time set in bit 31 to bit 16 when RX FIFO free space is lower than the low threshold.<br>The unit is 128 bytes. Default value is 4'h4. |
| [7:5] | - | - | - | Reserved |
| 4 | RC | 1'h0 | RX_PAUSE | Receives pause frame |
| 3 | RO | 1'h0 | TXPAUSED | Packet transmission is paused due to receive pause frame. |
| 2 | R/W | 1'h0 | FCTHR_EN | Enables flow control threshold mode<br>Enables transmit pause frame for high/low threshold. |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| 1 | R/W | 1'h0 | TX_PAUSE | Transmit pause frame<br>Software can set this bit to send pause frame. Auto cleared after pause frame has been transmitted. |
| 0 | R/W | 1'h0 | FC_EN | Flow control mode enable |

### 16.5.16 Back Pressure Register, BPR, 32'h0000_0400, (Offset: 0x9C ~ 0x9F)

**Table 16-21. Back Pressure Register, BPR, 32'h0000_0400, (Offset: 0x9C ~ 0x9F)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:8] | R/W | 4'h4 | BK_LOW | RX FIFO free space low threshold<br>MAC would generate the jam pattern when RX FIFO free space is lower than the low threshold when packets are incoming.<br>The unit is 128 bytes. Default value is 4'h4. |
| [7:4] | R/W | 4'h0 | BKJAM_LEN | Back pressure jam length<br>4'h0: 4 bytes 4'h6: 256 bytes<br>4'h1: 8 bytes 4'h7: 512 bytes<br>4'h2: 16 bytes 4'h8: 1024 bytes<br>4'h3: 32 bytes 4'h9: 1518 bytes<br>4'h4: 64 bytes 4'hA: 2048 bytes<br>4'h5: 128 bytes else: 4 bytes |
| [3:2] | - | - | - | Reserved |
| 1 | R/W | 1'h0 | BK_MODE | Back pressure address mode<br>1: Generate jam pattern when packet address matches.<br>0: Generate jam pattern when any packet is incoming. |
| 0 | R/W | 1'h0 | BK_EN | Back pressure mode enable |

## 16.5.17 Wake-On-LAN Control Register, WOLCR, 32'h0, (Offset: 0xA0 ~ 0xA3)

**Table 16-22. Wake-On-LAN Control Register, WOLCR, 32'h0, (Offset: 0xA0 ~ 0xA3)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:24] | R/W | 2'h0 | WOL_TYPE | WOL output signal type<br>2'b00: Active high<br>2'b01: Active low<br>2'b10: Positive pulse<br>2'b11: Negative pulse |
| [23:19] | - | - | - | Reserved |
| 18 | R/W | 1'h0 | SW_PDNPHY | Software power-down PHY |
| [17:16] | R/W | 2'h0 | WAKEUP_SEL | Wake-up frame selection<br>It is used to select the wake-up frame when the driver accesses wake-up frame CRC or wake-up frame byte mask register.<br>2'b00: Select wake-up frame 1<br>2'b01: Select wake-up frame 2<br>2'b10: Select wake-up frame 3<br>2'b11: Select wake-up frame 4 |
| [15:14] | R/W | 2'h0 | POWER_STATE | Power state<br>This 2-bit field is used to determine the current power state. The definition of the field values is given below.<br>2'b00: D0<br>2'b01: D1<br>2'b10: D2<br>2'b11: D3 |
| [13:7] | - | - | - | Reserved |
| 6 | R/W | 1'h0 | WAKEUP4_EN | Wake-up frame 4 event enable |
| 5 | R/W | 1'h0 | WAKEUP3_EN | Wake-up frame 3 event enable |
| 4 | R/W | 1'h0 | WAKEUP2_EN | Wake-up frame 2 event enable |
| 3 | R/W | 1'h0 | WAKEUP1_EN | Wake-up frame 1 event enable |
| 2 | R/W | 1'h0 | MAGICPKT_EN | Magic packet event enable |
| 1 | R/W | 1'h0 | LINKCHG1_EN | Link change to 1 event enable |
| 0 | R/W | 1'h0 | LINKCHG0_EN | Link change to 0 event enable |

## 16.5.18 Wake-On-LAN Status Register, WOLSR, 32'h0, (Offset: 0xA4 ~ 0xA7)

**Table 16-23. Wake-On-LAN Status Register, WOLSR, 32'h0, (Offset: 0xA4 ~ 0xA7)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| 6 | R/W1C | 1'h0 | WAKEUP4_STS | Wake-up frame 4 event status |
| 5 | R/W1C | 1'h0 | WAKEUP3_STS | Wake-up frame 3 event status |
| 4 | R/W1C | 1'h0 | WAKEUP2_STS | Wake-up frame 2 event status |
| 3 | R/W1C | 1'h0 | WAKEUP1_STS | Wake-up frame 1 event status |
| 2 | R/W1C | 1'h0 | MAGICPKT_STS | Magic packet event status |
| 1 | R/W1C | 1'h0 | LINKCHG1_STS | Link change to 1 event status |
| 0 | R/W1C | 1'h0 | LINKCHG0_STS | Link change to 0 event status |

Note: WOLSR can only work in the power-saving modes (D1 ~ D3). In the normal mode (D0), it is meaningless.

## 16.5.19 Wake-up Frame CRC Register, WFCRC, 32'h0, (Offset: 0xA8 ~ 0xAB)

**Table 16-24. Wake-up Frame CRC Register, WFCRC, 32'h0, (Offset: 0xA8 ~ 0xAB)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | WFCRC | Wake-up frame CRC value<br>When WAKEUP_SEL = 2'b00, the write value is to be stored in wake-up frame 1 CRC register. The read value is from wake-up frame 1 CRC register.<br>When WAKEUP_SEL = 2'b01, the write value is to be stored in wake-up frame 2 CRC register. The read value is from wake-up frame 2 CRC register.<br>When WAKEUP_SEL = 2'b10, the write value is to be stored in wake-up frame 3 CRC register. The read value is from wake-up frame 3 CRC register.<br>When WAKEUP_SEL = 2'b11, the write value is to be stored in wake-up frame 4 CRC register. The read value is from wake-up frame 4 CRC register. |

### 16.5.20 Wake-up Frame Byte Mask 1st Double Word Register, WFBM1, 32'h0, (Offset: 0xB0 ~ 0xB3)

**Table 16-25. Wake-up Frame Byte Mask 1st Double Word Register, WFBM1, 32'h0, (Offset: 0xB0 ~ 0xB3)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | WFBM1 | Wake-up frame byte mask 1st double word<br><br>It stores the byte mask of the wake-up frame from byte 1 to byte 32. Bit 0 is byte 1 mask and Bit 31 is byte 32 mask.<br><br>When WAKEUP_SEL = 2'b00, the write value is to be stored in wake-up frame byte 1 mask of the first double-word register. The read value is from wake-up frame byte 1 mask of the first double-word register.<br><br>When WAKEUP_SEL = 2'b01, the write value is stored in the wake-up frame byte 2 mask of the first double--word register. The read value is from byte mask of the first double-word register of wake-up frame 2.<br><br>When WAKEUP_SEL = 2'b10, the write value is to be stored in the wake-up frame byte 3 mask of the first double-word register. The read value is from byte mask of the first double-word register of wake-up frame 3.<br><br>When WAKEUP_SEL = 2'b11, the write value is stored in the wake-up frame byte 4 mask of the first double-word register. The read value is from byte mask of the first double-word register of wake-up frame 4. |

### 16.5.21 Wake-up Frame Byte Mask 2nd Double Word Register, WFBM2, 32'h0, (Offset: 0xB4 ~ 0xB7)

**Table 16-26. Wake-up Frame Byte Mask 2nd Double Word Register, WFBM2, 32'h0, (Offset: 0xB4 ~ 0xB7)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | WFBM2 | Wake-up frame byte mask 2nd double word<br><br>It stores the byte mask of the wake-up frame from byte 33 to byte 64. Bit 0 is byte 33 mask and bit 31 is byte 64 mask.<br><br>When WAKEUP_SEL = 2'b00, the write value is stored in the wake-up frame byte 1 mask of the second double-word register. The read value is from byte mask of the second double-word register of wake-up frame 1.<br><br>When WAKEUP_SEL = 2'b01, the write value is stored in the wake-up frame byte 2 mask of the second double-word register. The read value is from byte mask of the second double-word register of wake-up frame 2.<br><br>When WAKEUP_SEL = 2'b10, the write value is stored in the wake-up frame byte 3 mask of the second double-word register. The read value is from byte mask of the second double-word register of wake-up frame 3.<br><br>When WAKEUP_SEL = 2'b11, the write value is stored in the wake-up frame byte 4 mask of the second double-word register. The read value is from byte mask of the second double-word register of wake-up frame 4. |

### 16.5.22 Wake-up Frame Byte Mask 3rd Double Word Register, WFBM3, 32'h0, (Offset: 0xB8 ~ 0xBB)

**Table 16-27. Wake-up Frame Byte Mask 3rd Double Word Register, WFBM3, 32'h0, (Offset: 0xB8 ~ 0xBB)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | WFBM3 | Wake-up frame byte mask of the third double–word register<br><br>It stores the byte mask of the wake-up frame from byte 65 to byte 96. Bit 0 is byte 65 mask and Bit 31 is byte 96 mask.<br><br>When WAKEUP_SEL = 2'b00, the write value is stored in the wake-up frame byte 1 mask of the third double-word register. The read value is from byte mask of the third double-word register of wake-up frame 1.<br><br>When WAKEUP_SEL = 2'b01, the write value is stored in the wake-up frame byte 2 mask of the third double-word register. The read value is from byte mask of the third double-word register of wake-up frame 2.<br><br>When WAKEUP_SEL = 2'b10, the write value is stored in the wake-up frame byte 3 mask of the third double-word register. The read value is from byte mask of the third double-word register of wake-up frame 3.<br><br>When WAKEUP_SEL = 2'b11, the write value is stored in the wake-up frame byte 4 mask of the third double-word register. The read value is from byte mask of the third double-word register of wake-up frame 4. |

### 16.5.23 Wake-up Frame Byte Mask 4th Double Word Register, WFBM4, 32'h0, (Offset: 0xBC ~ 0xBF)

**Table 16-28. Wake-up Frame Byte Mask 4th Double Word Register, WFBM4, 32'h0, (Offset: 0xBC ~ 0xBF)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | R/W | 32'h0 | WFBM4 | Wake-up frame byte mask of the forth double–word register<br>It stores the byte mask of the wake-up frame from byte 97 to byte 128. Bit 0 is byte 97 mask and bit 31 is byte 128 mask.<br>When WAKEUP_SEL = 2'b00, the write value is stored in the wake-up frame byte 1 mask of the forth double-word register. The read value is from byte mask of the forth double-word register of wake-up frame 1.<br>When WAKEUP_SEL = 2'b01, the write value is stored in the wake-up frame byte 2 mask of the forth double-word register. The read value is from byte mask of the forth double-word register of wake-up frame 2.<br>When WAKEUP_SEL = 2'b10, the write value is stored in the wake-up frame byte 3 mask of the forth double-word register. The read value is from byte mask of the forth double-word register of wake-up frame 3.<br>When WAKEUP_SEL = 2'b11, the write value is stored in the wake-up frame byte 4 mask of the forth double-word register. The read value is from byte mask of the forth double-word register of wake-up frame 4. |

### 16.5.24 Test Seed Register, TS, 32'h0, (Offset: 0xC4 ~ 0xC7)

**Table 16-29. Test Seed Register, TS, 32'h0, (Offset: 0xC4 ~ 0xC7)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:14] | - | - | - | Reserved |
| [13:0] | R/W | 14'h0 | Test_seed | Test seed |

### 16.5.25 DMA/FIFO State Register, DMAFIFOS, 32'h0, (Offset: 0xC8 ~ 0xCB)

**Table 16-30. DMA/FIFO State Register, DMAFIFOS, 32'h0, (Offset: 0xC8 ~ 0xCB)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| 31 | RO | 1'h0 | TXD_REQ | TXDMA request |
| 30 | RO | 1'h0 | RXD_REQ | RXDMA request |
| 29 | RO | 1'h0 | DARB_TXGNT | TXDMA grant |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| 28 | RO | 1'h0 | DARB_RXGNT | RXDMA grant |
| 27 | RO | 1'h1 | TXFIFO_EMPTY | TX FIFO is empty |
| 26 | RO | 1'h1 | RXFIFO_EMPTY | RX FIFO is empty |
| [25:15] | - | - | - | Reserved |
| [14:12] | RO | 3'h0 | TXDMA2_SM | TXDMA 2 state machine |
| [11:8] | RO | 4'h0 | TXDMA1_SM | TXDMA 1 state machine |
| 7 | - | - | - | Reserved |
| [6:4] | RO | 3'h0 | RXDMA2_SM | RXDMA 2 state machine |
| [3:0] | RO | 4'h0 | RXDMA1_SM | RXDMA 1 state machine |

## 16.5.26 Test Mode Register, TM, 32'h0, (Offset: 0xCC ~ 0xCF)

**Table 16-31. Test Mode Register, TM, 32'h0, (Offset: 0xCC ~ 0xCF)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| 29 | R/W | 1'h0 | BIG_FORM | Big form register test mode |
| 28 | R/W | 1'h0 | PKT_SWAP_B | Byte swap packet data inside FTMAC110 when FTMAC110 is in the big-endian mode. Low active. |
| 27 | R/W | 1'h0 | DMA_SWAP | Byte swap AHB bus data when FTMAC110 is in the big-endian mode. |
| 26 | R/W | 1'h0 | SINGLE_PKT | Single packet mode. The TXDMA would only move a packet into the TX FIFO at one time. |
| 25 | R/W | 1'h0 | PTIMER_TEST | Automatic polling timer test mode |
| 24 | R/W | 1'h0 | ITIMER_TEST | Interrupt timer test mode |
| 23 | - | - | - | Reserved |
| 22 | R/W | 1'h0 | TEST_SEED_SEL | Test seed selection<br>When set, the least significant 14 bits of MAC_LADR (0x0C ~ 0x0F) is used as the seed. When cleared, Test_seed in TS register (0xC4 ~ 0xC7) is used as seed. |

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| 21 | R/W | 1'h0 | SEED_SEL | Seed selection<br>When set, internal counter is used as the seed. When cleared, external data are used as the seed. |
| 20 | R/W | 1'h0 | TEST_MODE | Transmission test mode |
| [19:10] | R/W | 10'h0 | TEST_TIME | Transmission back off time test |
| [9:5] | R/W | 5'h0 | TEST_EXCEL | Excessive collision test for transmission |
| [4:0] | - | - | - | Reserved |

## 16.5.27 TX_MCOL and TX_SCOL Counter Register, 32'h0, (Offset: 0xD4 ~ 0xD7)

**Table 16-32. TX_MCOL and TX_SCOL Counter Register, 32'h0, (Offset: 0xD4 ~ 0xD7)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | RO | 16'h0 | TX_MCOL | Counter for counting packets transmitted OK with 2~ 15 collisions |
| [15:0] | RO | 16'h0 | TX_SCOL | Counter for counting packets transmitted OK with single collision |

## 16.5.28 RPF and AEP Counter Register, 32'h0, (Offset: 0xD8 ~ 0xDB)

**Table 16-33. RPF and AEP Counter Register, 32'h0, (Offset: 0xD8 ~ 0xDB)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | RO | 16'h0 | RPF | Receive pause frame counter |
| [15:0] | RO | 16'h0 | AEP | Counter for counting packets with alignment error<br>The counter is to count packets with CRC error and no-octet-boundary discarded by FTMAC110. |

## 16.5.29 XM and PG Counter Register, 32'h0, (Offset: 0xDC ~ 0xDF)

**Table 16-34. XM and PG Counter Register, 32'h0, (Offset: 0xDC ~ 0xDF)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | RO | 16'h0 | XM | Counter for counting packets failed in transmission (Due to late collision or collision count ≥ 16) |
| [15:0] | RO | 16'h0 | PG | Counter for counting packets failed in transmission (Due to collision count ≥16) |

### 16.5.30 RUNT_CNT and TLCC Counter Register, 32'h0, (Offset: 0xE0 ~ 0xE3)

**Table 16-35. RUNT_CNT and TLCC Counter Register, 32'h0, (Offset: 0xE0 ~ 0xE3)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | RO | 16'h0 | RUNT_CNT | Counter for counting received runt packets |
| [15:0] | RO | 16'h0 | TLCC | Late collision counter |

### 16.5.31 CRCER_CNT and FTL_CNT Counter Register, 32'h0, (Offset: 0xE4 ~ 0xE7)

**Table 16-36. CRCER_CNT and FTL_CNT Counter Register, 32'h0, (Offset: 0xE4 ~ 0xE7)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | RO | 16'h0 | CRCER_CNT | CRC error packet counter<br>The counter counts the number of octet-boundary frames discarded because of a CRC error. |
| [15:0] | RO | 16'h0 | FTL_CNT | Counter for counting received FTL (Packet length > 1518 bytes) packets |

### 16.5.32 RLC and RCC Counter Register, 32'h0, (Offset: 0xE8 ~ 0xEB)

**Table 16-37. RLC and RCC Counter Register, 32'h0, (Offset: 0xE8 ~ 0xEB)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:16] | RO | 16'h0 | RLC | Counter for counting loss of received packets (Due to RX FIFO full) |
| [15:0] | RO | 16'h0 | RCC | Receive collision counter |

### 16.5.33 BROC Counter Register, 32'h0, (Offset: 0xEC ~ 0xEF)

**Table 16-38. BROC Counter Register, 32'h0, (Offset: 0xEC ~ 0xEF)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | RO | | BROC | Counter for counting received broadcast packets |

### 16.5.34 MULCA Counter Register, 32'h0, (Offset: 0xF0 ~ 0xF3)

**Table 16-39. MULCA Counter Register, 32'h0, (Offset: 0xF0 ~ 0xF3)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | RO | - | MULCA | Counter for counting the received multicast packets |

### 16.5.35 RP Counter Register, 32'h0, (Offset: 0xF4 ~ 0xF7)

**Table 16-40. RP Counter Register, 32'h0, (Offset: 0xF4 ~ 0xF7)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | RO | 32'h0 | RP | Counter for counting the successfully received packets |

### 16.5.36 XP Counter Register, 32'h0, (Offset: 0xF8 ~ 0xFB)

**Table 16-41. XP Counter Register, 32'h0, (Offset: 0xF8 ~ 0xFB)**

| Bit | RW Type | Default Value | Name | Description |
|---|---|---|---|---|
| [31:0] | RO | - | XP | Counter for counting the successfully transmitted packets |

## 16.6 Function Description

### 16.6.1 Half-duplex (CSMA/CD Access Protocol)

Half-duplex Ethernet is a traditional form of Ethernet that uses the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol with two or more CSMA/CD stations sharing a common transmission medium. To transmit a frame, a station must wait for the idle period on the medium when no other station is transmitting. The station then transmits the frame by broadcasting it over the medium in order for it to be heard by all other stations on the network. A "collision" may occur if another device tries to send data at the same time. The transmitting station then intentionally transmits a "jam sequence" to ensure that all stations are notified of the frame transmission failure resulted from a collision. The station then remains silent for a random period of time before attempting to transmit again. This process is repeated until the frame is transmitted successfully.

The basic rules for transmitting a frame are as follows: The network is monitored for an active "carrier", or presence of a transmitting station. This process is known as "carrier sense". If an active carrier is detected, the transmission will be deferred. The station continues to monitor the network until the carrier ceases. If an active carrier is not detected, and the period of absence of carrier is equal to or greater than the inter-frame gap, the station will then immediately begin transmitting the frame.

If a collision is detected during sending the frame, the transmitting station will stop sending the frame data and sends a 32-bit "jam sequence." If the collision is detected during transmitting the frame preamble, the transmitting station will complete sending the frame preamble before starting transmitting the jam sequence. The jam sequence will be transmitted to ensure that the length of the collision is sufficient to be noticed by the other transmitting stations. After sending the jam sequence, the transmitting station will wait for a random period of time that is chosen using a random number generator before starting the transmission process. This process is called "back off".

Having the colliding stations wait for a random period of time before retransmitting will decrease the probability of a repeated collision. When repeated collisions occur, transmission will be repeated; however, the random delay will increase with each attempt. This will further reduce the probability of another collision. This process repeats until a station transmits a frame without collision. Once a station successfully transmits a frame, it will clear the collision counter used to increase the back off time after each repeated collision.

### 16.6.2 Full-duplex Ethernet

The IEEE 802.3x standard defines a second mode of operation for Ethernet, called "full-duplex", which bypasses the CSMA/CD protocol. The CSMA/CD protocol is "half-duplex". This implies that a station may either transmit data or receive data, but never does both at the same time. The full-duplex mode enables two stations to simultaneously exchange data over a point-to-point link that provides independent transmit and receive paths. Since each station can simultaneously transmit and receive data, the aggregate throughput of the link is effectively doubled. A 10-Mbps station operating in the full-duplex mode provides a maximum bandwidth of 20 Mbps. A full-duplex 100 Mbps station provides a bandwidth of 200 Mbps.

The full-duplex operation is restricted in links that meet the following criteria:

- The physical medium must be capable of supporting simultaneous transmission and reception without interference. The media specifications that meet this requirement are 10BASE-T, 10BASE-FL, 100BASE-TX, 100BASE-FX, 100BASE-T2, 1000BASE-CX, 1000BASE-SX, 1000BASE-LS, and 1000BASE-T. The following media specifications do not support the full–duplex operation: 10BASE5, 10BASE2, 10BASE-FP, 10BASE-FB, and 100BASET4.
- The full-duplex operation is restricted to the point-to-point links. Since there is no contention for shared media, collisions cannot occur and the CSMA/CD protocol is unnecessary. Frames may be transmitted back-to-back with the interval of the minimum inter-frame gap.
- Both stations on the link must be capable of, and be configured for, the full-duplex operation.

The full-duplex operation offers several major advantages:

- Throughput is doubled by permitting simultaneous transmission and reception. The efficiency of the link is improved by eliminating the potential for collisions.
- Segment lengths are no longer limited by the timing requirements of half-duplex Ethernet that ensure collisions are propagated to all stations within the required 512-bit times. For example, 100BASE-FX is limited to the 412-meter segment length in the half-duplex mode but may support the segment lengths as long as 2 km in the full-duplex mode.

### 16.6.3 Loop Back

When FTMAC110 is configured in the "Loop Back" mode, FTMAC110 will loop the transmit data back through the MII interface and FTMAC110 will receive the data it transmitted. Users can test the control circuit and data path in this mode.

### 16.6.4 Transmit Descriptors and Data Buffers

FTMAC110 uses the descriptor ring to manage the transmit buffers. The transmit descriptors and buffers are all in system memory. FTMAC110 moves the transmit packet data from the transmit buffers in system memory to the TX FIFO inside FTMAC110. FTMAC110 then transmits the packet to Ethernet. The transmit descriptors that reside in the system memory act as pointers to the transmit buffers.

There is one descriptor ring for transmission. The base address of the transmit ring is in the Transmit Ring Base Address Register (TXR_BADR, offset: 0x20 ~ 0x23). Each transmit descriptor contains a transmit buffer. A transmit buffer consists of either an entire frame or a part of a frame; however, it cannot exceed a single frame. The transmit descriptor contains the transmit buffer status and the transmit buffer can only contain the transmit data.

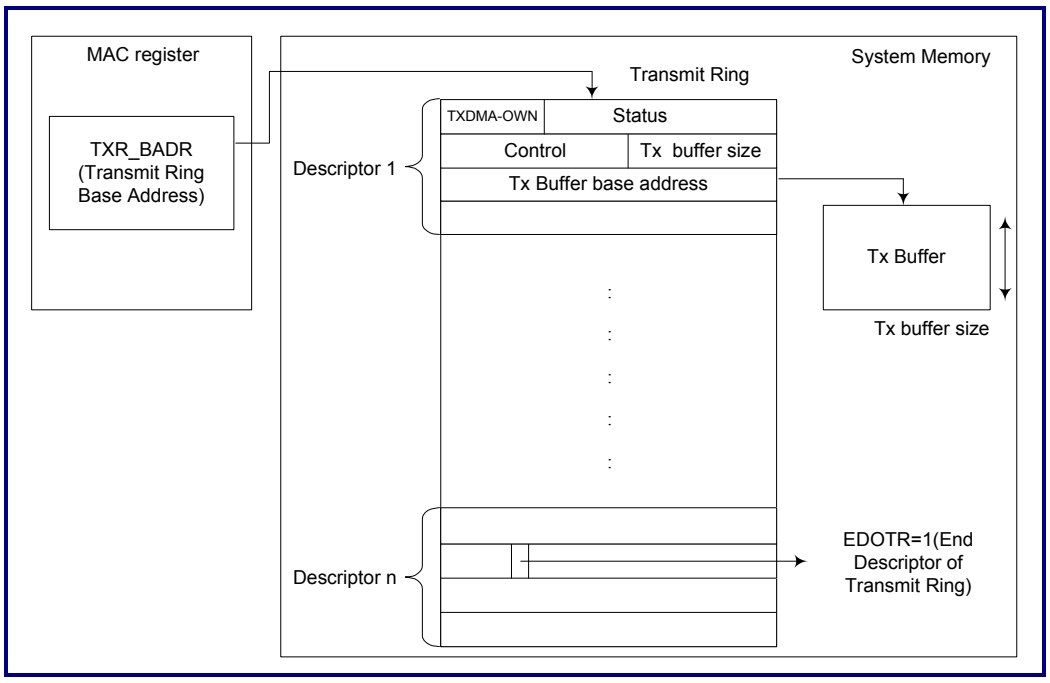



**Figure 16-1. Transmit Ring Descriptor Structure**

The transmit descriptor structure is as follows.

Notes:

1. The start address of each transmit descriptor must be 16-byte aligned.
2. SW must access transmit descriptor in WORD size when FTMAC110 is in big-endian mode.
   Limitation: The maximum transmit packet size is 2028 bytes.

| | | |
|---|---|---|
| TXDES0 | TXDMA_OWN | Status |
| TXDES1 | Control | TX buffer size |
| TXDES2 | TX buffer base address | |

TXDES0: It contains the transmit frame status and descriptor ownership information.

| Bit | Name | Description |
|---|---|---|
| 31 | TXDMA_OWN | TXDMA_OWN - TXDMA ownership bit<br>When this bit is set, it indicates that FTMAC110 owns the descriptor. When reset, this bit indicates that the software owns the descriptor. FTMAC110 clears this bit when it completes the frame transmission. |
| [30:2] | - | Reserved |
| 1 | TXPKT_EXSCOL | TXPKT_EXSCOL<br>When this bit is set, it indicates that the frame transmission is aborted after 16 collisions. It is valid only when FTS = 1. |
| 0 | TXPKT_LATECOL | TXPKT_LATECOL<br>When this bit is set, it indicates that the frame transmission is aborted due to late collision. It is valid only when FTS = 1. |

TXDES1: It contains the control bits and the transmit buffer size.

| Bit | Name | Description |
|---|---|---|
| 31 | EDOTR | EDOTR - End Descriptor of Transmit Ring<br>When this bit is set, it indicates that the descriptor is the last descriptor of the transmit ring. |
| 30 | TXIC | TXIC - Transmit Interrupt on Completion<br>When this bit is set, FTMAC110 will assert transmit interrupt after the present frame has been transmitted. This bit is valid only when FTS = 1 and when bit 8 to bit 14 (TXINT_THR, TXINT_CNT) of the Interrupt Timer Control Register = 0. |
| 29 | TX2FIC | TX2FIC - Transmit to FIFO Interrupt on Completion<br>When this bit is set, FTMAC110 will assert transmit interrupt after the present frame has been moved into the TX FIFO. This bit is valid only when FTS = 1. |
| 28 | FTS | FTS - First Transmit Segment descriptor<br>When this bit is set, it indicates that this is the first descriptor of a TX packet. |
| 27 | LTS | LTS - Last Transmit Segment descriptor<br>When this bit is set, it indicates that this is the last descriptor of a TX packet. |
| [26:11] | - | Reserved |

| Bit | Name | Description |
|---|---|---|
| [10:0] | TXBUF_SIZE | Transmit buffer size in byte<br>When the size is 0, the descriptor will be discarded. |

TXDES2: It contains the transmit buffer base address.

| Bit | Name | Description |
|---|---|---|
| [31:0] | TXBUF_BADR | Transmit buffer base address |

### 16.6.5 Receive Descriptors and Data Buffers

FTMAC110 implements a descriptor ring to manage the receive buffers. The receive descriptors and buffers are all in the system memory. FTMAC110 first stores the packet received from the network in the RX FIFO and then moves the received packet data to the receive buffers in system memory. The receive descriptors that reside in the system memory act as pointers to the receive buffers.

There is one descriptor ring for reception. The base address of the receive ring is in the Receive Ring Base Address Register (RXR_BADR, offset: 0x24 ~ 0x23). Each receive descriptor contains a receive buffer. A receive buffer consists of either an entire frame or part of a frame, but it cannot exceed a single frame. The receive descriptor contains receive buffer status and the receive buffer can only contain the receive packet data.

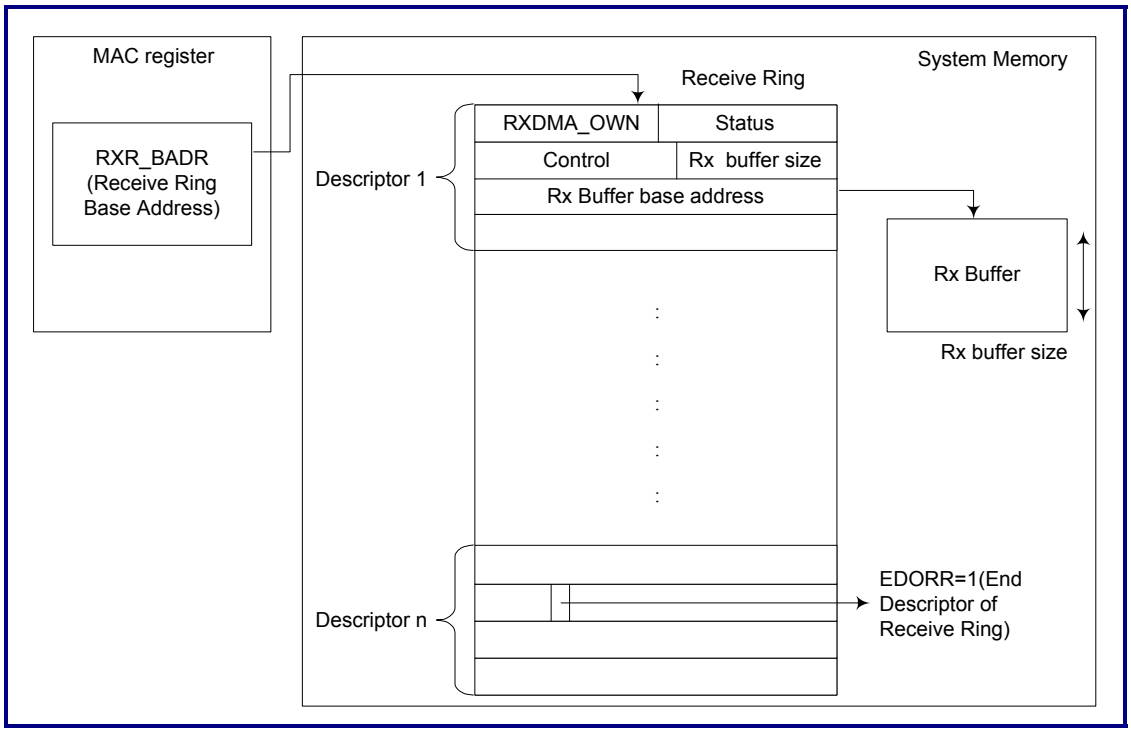



**Figure 16-2. Receive Ring Descriptor Structure**

The receive descriptor structure is as follows:

Notes:

1. The start address of each receive descriptor must be 16-byte aligned.
2. SW must access receive descriptor in WORD size when FTMAC110 is in big-endian mode.
   Limitation: The maximum receive packet size that FTMAC110 can receive is 2020 bytes.

| | | |
|---|---|---|
| RXDES0 | RXDMA_OWN | Status |
| RXDES1 | Control | RX buffer size (It should be in double word) |
| RXDES2 | RX buffer base address (It should be 4-byte-aligned) | |

RXDES0: It contains the receive frame status and descriptor ownership information.

| Bit | Name | Description |
|---|---|---|
| 31 | RXDMA_OWN | RXDMA_OWN - RXDMA ownership bit<br>When this bit is set, it indicates that the descriptor is owned by FTMAC110. When reset, it indicates that the descriptor is owned by the software. FTMAC110 clears this bit when it completes the frame reception or when the receive buffer of the receive descriptor is full. |
| 30 | - | Reserved |
| 29 | FRS | FRS - First Receive Segment descriptor<br>When this bit is set, it indicates that this is the first descriptor of a received packet. |
| 28 | LRS | LRS - Last Receive Segment descriptor<br>When this bit is set, it indicates that this is the last descriptor of a received packet. |
| [27:23] | - | Reserved |
| 22 | RX_ODD_NB | RX_ODD_NB - Receive Odd Nibbles<br>When this bit is set, it indicates receiving a packet with odd nibbles. |
| 21 | RUNT | RUNT - Runt packet<br>When this bit is set, it indicates that the received packet length is less than 64 bytes.<br>Bits[23:0] are valid only when the FRS = 1. |
| 20 | FTL | FTL - Frame Too Long<br>When this bit is set, it indicates that the received packet length exceeds 1518 bytes. |
| 19 | CRC_ERR | CRC_ERR - CRC error<br>When this bit is set, it indicates that the CRC error occurs on the received packet. |
| 18 | RX_ERR | RX_ERR - Receive error<br>When this bit is set, it indicates that receive error happens when receiving a packet. |
| 17 | BROADCAST | BROADCAST - Broadcast frame<br>When this bit is set, it indicates that the received packet is a broadcast frame. |
| 16 | MULTICAST | MULTICAST - Multicast frame<br>When this bit is set, it indicates that the received packet is a multicast frame. |
| [15:11] | - | Reserved |
| [10:0] | - | RFL - Receive Frame Length |

RXDES1: It contains the control bits and the receive buffer size.

| Bit | Name | Description |
|---|---|---|
| 31 | EDORR | EDORR - End Descriptor of Receive Ring<br>When this bit is set, it indicates that the descriptor is the last descriptor of the receive ring. |
| [30:11] | - | Reserved |
| [10:0] | RXBUF_SIZE | Receive buffer size<br>The unit is 1 byte. Receive buffer size must be double-word boundary (4-byte-aligned). |

RXDES2: It contains the receive buffer base address.

| Bit | Name | Description |
|---|---|---|
| [31:0] | RXBUF_BADR | Receive buffer base address<br>Receive buffer base address must be double-word boundary (4-byte-aligned). |

### 16.6.6 Transmitting Packets

When the software wants to transmit packet to Ethernet, it will first move the packet data into the transmit buffer. The software will then write the packet's length and position into the transmit descriptor and trigger FTMAC110 to send the packet. After the entire packet has been moved into the TX FIFO, FTMAC110 begins to transmit it to Ethernet. When the packet is transmitted; FTMAC110 will assert the interrupt to notify software that the packet has been transmitted successfully.

### 16.6.7 Receiving Packets

Whenever encountering an incoming packet, FTMAC110 will save the received packet in the RX FIFO if both the CRC result and address check result are correct. After the incoming packet is successfully saved in RX FIFO, FTMAC110 will initiate the Direct Memory Access (DMA) function to move the received packet data from the RX FIFO to the system memory. FTMAC110 then will assert the interrupt to notify software that the packet has been successfully received.

### 16.6.8 Zero-Copy

With the zero-copy function of FTMAC110, the system will not perform data movement for packet header alignment. DMA of FTMAC110 will place the first segment of the incoming packet at 2-byte-aligned address regardless of the assigned buffer address in the receive descriptor. If there are more segments (Packet size > receive buffer size), they will be placed from 4-byte-aligned address. Figure 3-3 shows an example of packet placement in a little-endian system. The system can determine the start address of valid data by the FRS flag in receive descriptor.

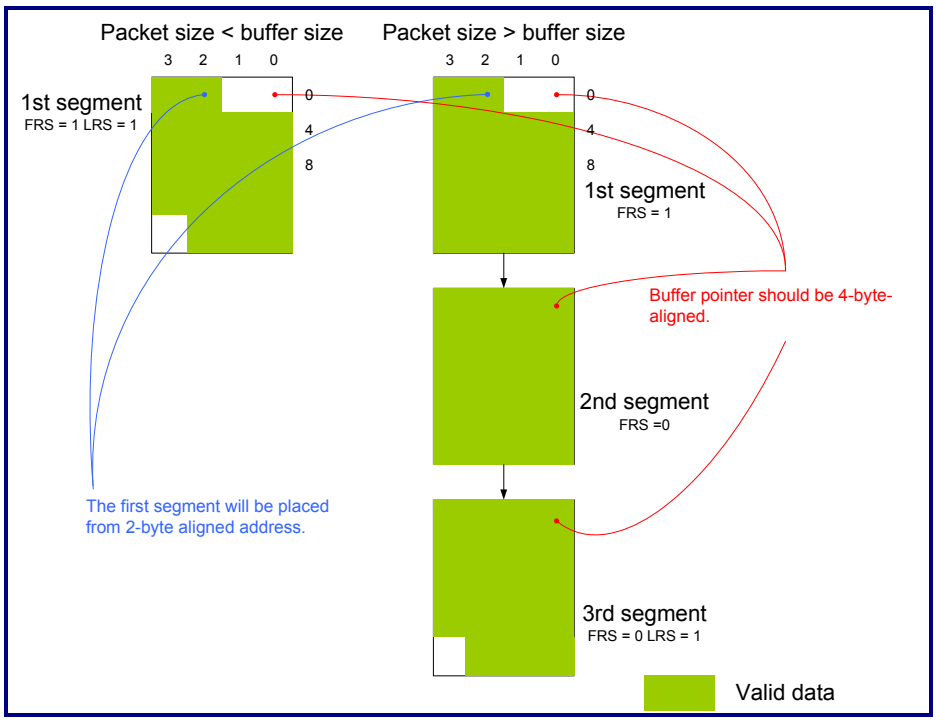



**Figure 16-3. Example of Incoming Packet Placement**

### 16.6.9 Ethernet Address Filtering

FTMAC110 can be set up to recognize any one of the Ethernet receive address groups described in the following table.

RX_BROADPKT: Bit 17 of MAC Control Register (Offset: 0x88)

RX_MULTIPKT: Bit 16 of MAC Control Register (Offset: 0x88)

RCV_ALL: Bit 12 of MAC Control Register (Offset: 0x88)

HT_MULTI_EN: Bit 9 of MAC Control Register (Offset: 0x88)

| RCV_ ALL | RX_ MULTIPKT | RX_ BROADPKT | HT_ MULTI_EN | Group | Description |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | A | FTMAC110 receives the following frame:<br>The destination address of a frame exactly matches the MAC_ADR (Offset: 0x08 ~ 0x0F) of FTMAC110. |
| 0 | 0 | 0 | 1 | B | FTMAC110 receives the following frames:<br>The destination address of a frame matches the MAC_ADR (Offset: 0x08 ~ 0x0F) of FTMAC110 exactly.<br>The destination address of a frame is a multicast address. Pass the address filtering of the multicast address hash table in FTMAC110. |
| 0 | 0 | 1 | 0 | C | FTMAC110 receives the following frames:<br>The destination address of a frame matches the MAC_ADR (Offset: 0x08 ~ 0x0F) of FTMAC110 exactly.<br>The destination address of a frame is a broadcast address. |
| 0 | 0 | 1 | 1 | D | FTMAC110 receives the following frames:<br>The destination address of a frame matches the MAC_ADR (Offset: 0x08 ~ 0x0F) of FTMAC110 exactly.<br>The frame's destination address is a multicast address. Pass the address filtering of the multicast address hash table in FTMAC110.<br>The frame's destination address is a broadcast address. |
| 0 | 1 | X | X | E | FTMAC110 receives the following frames:<br>The destination address of a frame matches the MAC_ADR (Offset: 0x08 ~ 0x0F) of FTMAC110 exactly.<br>The destination address of a frame is a multicast address. |
| 1 | X | X | X | F | FTMAC110 supports reception of all frames on the network regardless of their destination address. |

### 16.6.10 DMA Arbitration Scheme

The DMA arbitration scheme is decided by RX_THR_EN (Bit 9 of DMA Burst Length and Arbitration Control Register, offset: 0x30 ~ 0x33). When RX_THR_EN = 0, the DMA arbitration scheme will perform a fair arbitration between the TXDMA and RXDMA. The last one using the DMA channel has lower priority to get the DMA channel when the DMA channel are requested by both TXDMA and RXDMA at the same time.

When RX_THR_EN is set, if the space used in the RX FIFO is larger than or equal to RXFIFO_HTHR (Bits[8:6] of DMA Burst Length and Arbitration Control Register, offset: 0x30 ~ 0x33), the RXDMA has higher priority than TXDMA in using the DMA channel. RXDMA will keep the higher priority until the used space in the RX FIFO is less than or equal to RXFIFO_LTHR (Bits[5:3] of DMA Burst Length and Arbitration Control Register, offset: 0x30 ~ 0x33). The TXDMA will then get higher priority than RXDMA. Consequently, software must set RXFIFO_HTHR to be larger than RXFIFO_LTHR to keep FTMAC110 working correctly.

### 16.6.11 Wake-On-LAN

FTMAC110 supports the Wake-On-LAN function. The Wake-On-LAN function supports three wake-up events: Link status change, magic packet, and wake-up frame.

### 16.6.12 Link Status Change

Link status change refers to the event where the link state to Ethernet changes. PHY will offer a phy_linksts signal. If the link state to Ethernet changes, the state of phy_linksts will also change.

Once FTMAC110 has been put into the power-saving mode and the link status change mode is enabled, the link status change will be treated as a wake-up event.

### 16.6.13 Magic Packet

A magic packet contains a specific sequence consisting of 16 duplications of the node address of the network adaptor without breaks. This specific sequence must be preceded by 6 bytes of FFh. The format of a magic packet is as follows:

**DA + SA + … + 6 * (8'hFFh) + … + 16 * (network adaptor's node address) + …**

Once FTMAC110 is put into the power-saving mode and once the magic packet mode is enabled, a magic packet will be treated as a wake-up event.

### 16.6.14 Wake-up Frame

The purpose of the wake-up frame is to wake up the system when another machine on the network needs to communicate with this system. It does not require the application running on the remote machine to send a special wake-up frame pattern. Instead, when FTMAC110 is in the wake-up frame mode, it tries to identify certain interesting frames that are sent by existing network protocols. Some examples are NETBIOS name lookups and ARP requests.

Before putting FTMAC110 into the wake-up frame mode, the system will pass to the driver a list of wake-up frames that can wake up the system; and the driver will pass the information to FTMAC110 by writing the corresponding register. FTMAC110 will then identify if a packet is a wake-up frame according to the information.

Before putting FTMAC110 into the wake-up frame mode, the system should pass to the driver a list of wake-up frames and corresponding byte masks. Each byte mask defines which bytes of the incoming frames should be compared with the corresponding wake-up frame in order to determine whether or not to accept the incoming frames as a wake-up event.

**Table 16-42. Wake-up Frame Format**

| Wake-up Frame Format | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byte content | 00 | 25 | 67 | 90 | 44 | A3 | 6C | … |
| Byte mask | 0 | 0 | 0 | 1 | 1 | 0 | 1 | … |

There are two ways to identify if the received packet is a wake-up frame or not. Please note that FTMAC110 adopts the "Signature matching" approach.

- Exact matching: The approach of which FTMAC110 will need many registers to store all byte content and byte mask for each wake-up frame. When a packet arrives from the network, FTMAC110 will check the bytes of the incoming frame that correspond to the bits that are set to '1' in the byte mask for each wake-up frame. If the check result is OK for any wake-up frame and if the incoming frame passes the standard CRC check, FTMAC110 will treat it as a wake-up event.
- Signature matching: The approach of which FTMAC110 will need a CRC generation circuit and registers to save all byte mask and 4-byte CRC register for each wake-up frame. The driver calculates a CRC value based on those bytes of the wake-up frame that correspond to the bits that are set to '1' in the byte mask. The driver will store the resulting value and corresponding byte mask into FTMAC110. When the wake-up frame mode is enabled as a frame arrives from the network, each CRC generator will calculate a CRC value based on those bytes of the incoming frame that correspond to the bits that are set to '1' in that CRC generator byte mask. If the calculated value matches the stored value for any wake-up frame and if the incoming frame passes the standard CRC check, FTMAC110 will treat it as a wake-up event.

### 16.6.15 Power-down Mode

FTMAC110 has one power-down mode that significantly reduces power dissipation when its power state is not programmed into the D0 power state[5].

The features of the power-down mode are briefly listed as below: FTMAC110 will not assert an interrupt in the power-down mode.

- FTMAC110 will not transmit packets to Ethernet.
- FTMAC110 will not save received packets in the RX FIFO.
- FTMAC110 will assert WOL when a wake-up event happens.

### 16.6.16 Flow Control

FTMAC110 implements the flow control functionIt supports the IEEE802.3x flow control for full-duplex mode and backpressure for half-duplex mode.

The IEEE802.3x flow control is used in the full-duplex mode. When A and B are transmitting and receiving packet with each other in the full-duplex mode, if the RX FIFO in B is nearly full, B will send a pause frame to A in order to avoid packet loss. Then A will be inhibited from transmitting packet for a specified period of time while B is consuming the received data. A will continue to send packet to B after the pause time has lapsed. Brief features of the flow control in the full-duplex mode are as follows:

- The software is able to configure the pause time of the pause frame.
- FTMAC110 is able to send the pause frame according to the low/high threshold of RX FIFO.
- The software is able to send the pause frame by writing the register.

[5] The power-state representation of FTMAC110 follows the device state definition of ACPI (Advanced Configuration and Power Interface). D0 *Fully-ON* is the operating state. D3 *Off* powers off the device and will not response to its bus. D1 and D2 are the intermediate power states, whose definitions are varied by device types. In FTMAC110, these two states are the same as D3.

The back pressure mode is used in the half-duplex mode. When A is the transmitting and receiving packets in the half-duplex mode, if RX FIFO in A is nearly full, A will send a jam pattern to avoid packet from being saved into the RX FIFO, which allows A to consume the received data during that period of time. No jam pattern will be issued again and the receiving capability of RX FIFO will then be restored. Brief features of the back pressure mode are as follows:

- The software is able to configure the length of the jam.
- FTMAC110 is able to send the jam according to the low/high threshold of RX FIFO.

## 16.7 Initialization/Application Information

### 16.7.1 Frame Transmitting Procedure

The frame transmitting procedure is as follows:

Initialization:

1. Allocate system memory for the transmit descriptor ring and transmit buffer
2. Initialize the transmit descriptor ring
3. Set the Transmit Ring Base Address Register (Offset: 0x20 ~ 0x23) to the base address of the transmit descriptor ring in the system memory
4. Set the Interrupt Mask Register (Offset: 0x04 ~ 0x07)
5. Set the MAC Address Register (Offset: 0x08 ~ 0x0F)
6. Set the Multicast Address Hash Table Register (Offset: 0x10 ~ 0x17)
7. Set the Interrupt Timer Control Register (Offset: 0x28 ~ 0x2B) to select the manner of the transmit interrupt
8. Set the Automatic Polling Timer Control Register (Offset: 0x2C ~ 0x2F) to select the manner of transmitting poll
9. Set the MAC Control Register (Offset: 0x88 ~ 0x8B) to set valid configuration for FTMAC110 and enable transmit channel

Transmit procedures:

1. The software will check whether the remaining transmit descriptors is sufficient for the next packet transmission. If not, the software will wait until there are sufficient transmit descriptors.
2. Prepare the transmit packet data to the transmit buffer
3. Set the transmit descriptor
4. Write the Transmit Poll Demand Register (Offset: 0x18 ~ 0x1B) to trigger FTMAC110 to poll the transmit descriptor if necessary
5. Wait for interrupt.
6. When interrupt occurs, the software will check whether it is a transmit interrupt. If ISR[4] = 1, it means that the packet has been successfully transmitted to the network. If ISR[5] = 1, it means that the packet has been aborted during transmission due to late collision or excessive collision.

Notes:

- When setting the transmit descriptor, TXDES0 must be set last. Thus, the setting procedure should be one of the following procedures:
  - Procedure A
    - Set TXDES2
    - Set TXDES1
    - Set TXDES0
  - Procedure B
    - Set TXDES1
    - Set TXDES2
    - Set TXDES0
- When preparing a transmit packet which contains more than one transmit descriptor, the first transmit descriptor must be the last set descriptor of the transmit packet.

### 16.7.2 Frame Receiving Procedure

The receiving frame procedure is as follows:

Initialization:

1. Allocate system memory for the receive descriptor ring and receive buffer
2. Initialize the receive descriptor ring
3. Set the Receive Ring Base Address Register (Offset: 0x24 ~ 0x27) to the base address of the receive descriptor ring in the system memory
4. Set the Interrupt Mask Register (Offset: 0x04 ~ 0x07)
5. Set the MAC Address Register (Offset: 0x08 ~ 0x0F)
6. Set the Multicast Address Hash Table Register (Offset: 0x10 ~ 0x17)
7. Set the Interrupt Timer Control Register (Offset: 0x28 ~ 0x2B) to select the manner of the receive interrupt
8. Set the Automatic Polling Timer Control Register (Offset: 0x2C ~ 0x2F) to select the manner of receive poll
9. Set the MAC Control Register (Offset: 0x88 ~ 0x8B) to set valid configuration for FTMAC110 and enable receive channel
10. Write the Receive Poll Demand Register (Offset: 0x1C ~ 0x1F) to trigger FTMAC110 to poll the receive descriptor

Receive procedures:

1. Wait for interrupt
2. When interrupt occurs, the software will check whether it is a receive interrupt. If ISR[0] = 1, it means that the packet has been moved to the receive buffer successfully. Then software needs to fetch the receive descriptor to get the received packet until the owner bit of the next receive descriptor does not belong to the software.
3. The software needs to release the receive descriptors to FTMAC110 after accessing the received packet.
4. If the receive automatic poll function is disabled, software needs to write the Receive Poll Demand Register (Offset: 0x1C ~ 0x1F) to trigger FTMAC110 to poll the receive descriptor.

### 16.7.3 Procedures of Entering to and Exiting from Power-down Mode

The followings are the procedures to enter the power-down mode:

1. Set XDMA_EN (0x88.0) = 0
2. Poll the DMA/FIFO State Register (0xC8) to wait for TX FIFO empty
3. Set XMT_EN (0x88.5) = 0 to stop the transmission
4. Set RCV_EN (0x88.8) = 0
5. Poll the DMA/FIFO State Register (0xC8) to wait for RX FIFO empty
6. Set RDMA_EN (0x88.0) = 0 to stop reception
7. Program the Wake-up Frame CRC Register (0xA8), Wake-up Frame Byte Mask 1st Double Word Register (0xB0), Wake-up Frame Byte Mask 2nd Double Word Register (0xB4), Wake-up Frame Byte Mask 3rd Double Word Register (0xB8), and Wake-up Frame Byte Mask 4th Double Word Register (0xBC), if software wants to support the wake-up frame event in the power-saving mode.
8. Write 32'hFFFF_FFFF to clear the Wake-On-LAN Status Register (0xA4)
9. Program the requested wake-up events and power state into the Wake-On-LAN Register (0xA0) to let FTMAC110 go into the power-saving mode
10. Set RCV_EN (0x88.8) = 1 to enable reception

Procedures to exit the power-down mode:

1. Wait for the occurrence of wake-up events
2. Set RCV_EN (0x88.8) = 0 to stop reception
3. Read the Wake-On-LAN Status Register (0xA4) to check which wake-up event happened
4. Program the Wake-On-LAN Register (0xA0) to let FTMAC110 exit the power-saving mode
5. Set SW_RST (0x88.2) = 1 to reset FTMAC110
6. Check SW_RST (0x88.2) = 0 to make sure that FTMAC110 has finished reset
7. Re-initialize FTMAC110 to transmit and receive packets

### 16.7.4 Ethernet Frame Formats

**Table 16-43. FTMAC110 Frame Structure and Frame Formats**

| Preamble | SFD | DA | SA | LEN | LLC | PAD | CRC |
|---|---|---|---|---|---|---|---|
| 7-byte | 1-byte | 6-byte | 6-byte | 2-byte | 0-1500 byte | 0-46 byte | 4-byte |

- Preamble - 7 identical bytes
  The bits in each byte are '10101010' transmitted from left to right. The preamble is not required for reception. A sequence of 56 bits having alternating 1 and 0 values are used for synchronization. They serve to give components in the network time to detect the presence of a signal, and read the signal before the frame data arrives.
- Start of Frame Delimiter (SFD) - 1 byte
  The bits are '10101011' transmitted from left to right, which are required for reception.
- Destination Address (DA) - 6 bytes
  It may be an individual or a multicast or a broadcast address. The Destination MAC Address field identifies the station(s) to receive the frame. The Source MAC Address identifies the station that originated the frame. The 802.3 standard permits these address fields to be either 2-byte or 6-byte in length, but virtually all Ethernet implementations today use 6-byte addresses. A Destination Address may specify either an "individual address" destined for a single station, or a "multicast address" destined for a group of stations. A Destination Address of all 1 bits refers to all stations on the LAN and is called a "broadcast address."
- Source Address (SA) - 6 bytes
  The Source MAC Address identifies the station that originated the frame.
- Length (LEN) - 2 bytes
  Length values indicate the number of Logical Link Control (LLC) data bytes in the data field. If the value of this field is less than or equal to 1500, then the Length/Type field indicates the number of bytes in the subsequent MAC Data field. If the value of this field is greater than or equal to 1536, then the Length/Type field indicates the nature of the MAC protocol (Protocol type).
- Logical Link Control (LLC) Data - The user data.
- PAD - to 46 bytes. If the user data (From DA) is less than 46 bytes in length, PAD will be needed.

- Cyclic Redundancy Check (CRC) - four bytes. A value computed as a function of all fields except the preamble, the SFD, and the CRC itself. This field contains a 4-byte cyclical redundancy check (CRC) value used for error checking. When a source station assembles a MAC frame, it performs a CRC calculation on all the bits in the frame from the Destination MAC Address through the Pad fields (That is, all fields except the preamble, start of frame delimiter, and frame check sequence). The source station stores the value in this field and transmits it as a part of the frame. When the destination station receives the frame, it performs an identical check. If the calculated value does not match the value in this field, the destination station assumes an error has occurred during transmission and discards the frame.
- Interframe Gap - Ethernet devices must allow a minimum idle period between transmission of frames known as the interframe gap (IFG) or the inter-packet gap (IPG). It provides a brief recovery time between frames to allow devices to prepare for reception of the next frame. The minimum interframe gap is 96 bit times, which is 9.6 microseconds for 10 Mbps Ethernet, 0.96 microseconds for 100 Mbps Ethernet, and 0.096 microseconds for 1 Gbps Ethernet.
- The transmission order in the MII interface is Preamble → SFD → DA → SA → LEN → LLC → PAD (If necessary) →CRC. The most significant byte of each item is transmitted first. The most significant bit of each byte is transmitted first.

## 16.8 MII Management Interface

FTMAC110 contains an MII management interface for an MII compliant PHY device. This will pass the control and status parameters to FTMAC110 and PHY by MDIO and MDC, thereby reducing the number of control pins required for the PHY mode control. The protocol consists of the bit stream that is sampled at rising edge of the MDC; the bit stream format is described below.

**Table 16-44. PHY Bit Stream Format**

| | PRE | ST | OP | PHYAD | REGAD | TA | DATA | IDLE |
|---|---|---|---|---|---|---|---|---|
| READ | 1…1 | 01 | 10 | AAAAA | RRRRR | Z0 | D..D (16) | Z |
| WRITE | 1…1 | 01 | 01 | AAAAA | RRRRR | 10 | D..D (16) | Z |

- PRE (Preamble) - At the beginning of each transaction, the FTMAC110 sends a sequence of 32 contiguous logic bits on MDIO with 32 corresponding cycles on MDC to establish synchronization
- ST (Start of Frame) - A start of frame with 01
- OP (Operation Code) - 10 denotes read; 01 denotes write
- PHYAD (PHY Address) - A 5-bit address of PHY device. The first bit transmitted is MSB
- REGAD (Register Address) - A 5-bit address of PHY register. The first bit transmitted is MSB
- TA (Turn Around) - The turn around is a 2-bit time to avoid contention during a transaction
- DATA (Data) - The data field is 16-bit
- IDLE (Idle) - The condition on MDIO is high impedance state

## 16.9 Multicast Address Hash Table Filtering

There is a 64-bit multicast address hash table in FTMAC110. This hash table is used to filter the incoming packet with a multicast destination address. For an incoming frame with a multicast destination address, FTMAC110 uses the destination address (The first six bytes of the received packet) of a packet and the standard Ethernet Cyclic Redundancy Check (CRC) function to calculate the value of 32-bit CRC. FTMAC110 then uses the most significant six bits of the CRC value as the bit address index for the multicast address hash table. If the indexed bit in the table is set, the frame will pass the address filtering of the multicast address hash table. If the indexed bit in the table is set, the frame will pass the address filtering of the multicast address hash table. If the indexed bit in the table is cleared, the frame will not pass the address filtering of the multicast address hash table. If HT_MULTI_EN = 1 (Bit 9 of the MAC Control Register, offset: 0x88) and the frame passes the address filtering of the multicast address hash table, the frame will be accepted; otherwise the frame will be rejected.

Suppose that the destination address of incoming packet destination address is RX_DA = {RXDA5[7:0], RXDA4[7:0], RXDA3[7:0], RXDA2[7:0], RXDA1[7:0], RXDA0[7:0]}, the following provides an example of Verilog HDL that generates the address index to multicast address hash table for RX_DA.

```
module GEN_ADDRESS()
parameter CRC_POLY = 32'h04C1_1DB7;    // CRC poly
parameter RXPKT_DA = 32'h0102_0304_0506; // Receive packet's Destination address
integer I;
reg [47:0] RX_DA
reg [47:0] DATA_IN;
reg [31:0] CRC32;
reg [5:0] MAHT_ADR;
reg     NEW_BIT;

initial
begin
    RX_DA = RXPKT_DA;
DATA_IN = {RXDA0[7:0],RXDA1[7:0],RXDA2[7:0],RXDA3[7:0],RXDA4[7:0],RXDA5[7:0]};
CRC32 = 32'hFFFF_FFFF_FFFF;
for(I = 0; I<48; I = I+1)
begin
  NEW_BIT = CRC32[31] ^ DATA_IN[I];
  CRC32    = {CRC32[30:0], 1'b0}^{ {32{NEW_BIT}} & CRC_POLY };
end
MAHT_ADR =  CRC32[31:27];
End
endmodule
```

# Chapter 17

# TV Controller

This chapter contains the following sections:


- 17.1 General Description
- 17.2 Features
- 17.3 Block Diagram
- 17.4 Programming Model

## 17.1 General Description

FTTVE100 is an integrated digital video encoder. It converts the digital ITU-R BT656 YCbCr 4:2:2 video format to Composite, Y/C (S-video), and component YUV digital video output. The video output can be programmed to be compatible with NTSC-M, NTSC-J, PAL-B, D, G, H, I, M, N, and Combination N systems. FTTVE100 requires a single 27-MHz clock input to support all types of the video formats.

## 17.2 Features

**TV Interface**

- Provides Composite, Y/C (S-Video), and YUV component digital video outputs
- NTSC-M, NTSC-J, PAL-B, D, G, H, I, M, N, and combination N encoding
- On-chip color-bar generator

**Input Data Mode**

- BT601/656 pixel data

## 17.3 Block Diagram

The block diagram of the Faraday FTTVE100 controller is shown in Figure 17-1.

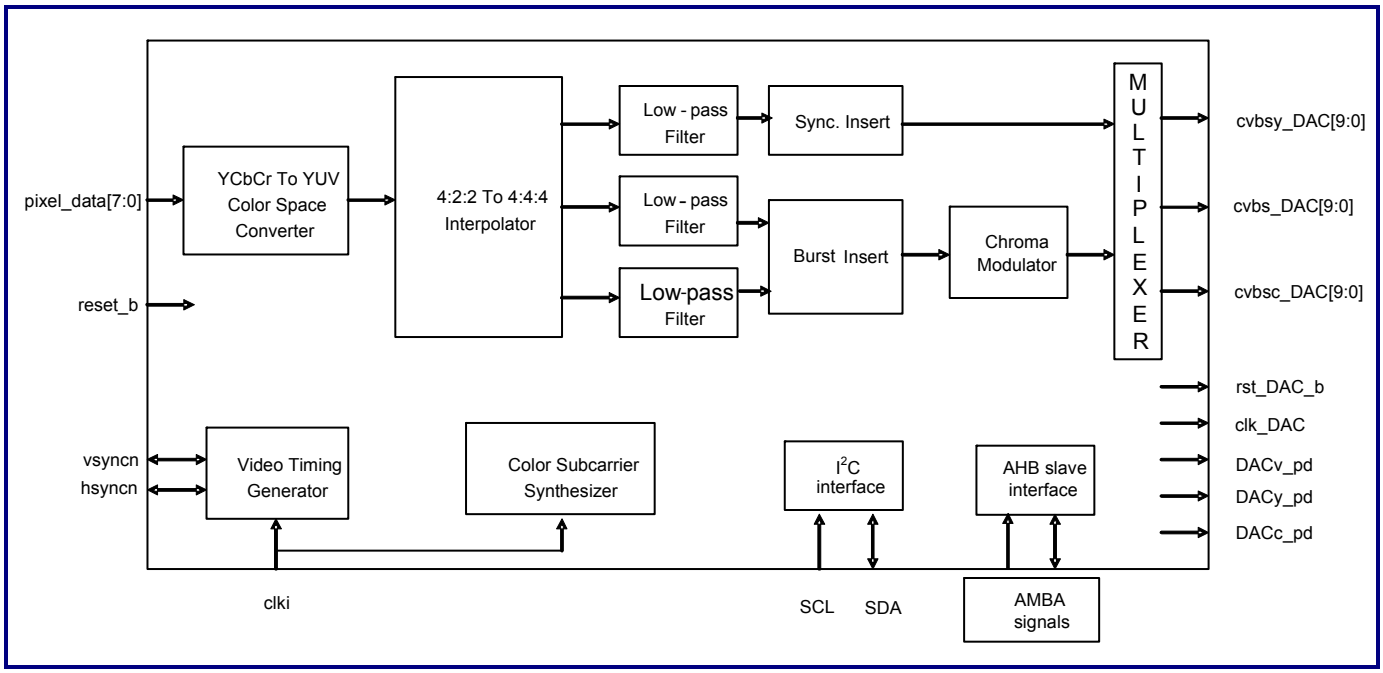



**Figure 17-1. Block Diagram of FTTVE100 Controller**

## 17.4 Programming Model

### 17.4.1 Summary of FTTVE100_S Controller Registers

**Table 17-1. FTTVE100_S Controller Register Summary**

| Address Offset | Type | Description | Reset Value |
|---|---|---|---|
| 0x0 | R/W | TVE Video Format Control Register | 0x0 |
| 0x4 | R/W | TVE Video Input Control Register | 0x0 |
| 0x8 | R/W | TVE Video Output Control Register | 0x0 |
| 0xC | R/W | TVE Power-down Control Register | 0x0 |
| 0x10 | W | TVE Software Reset Control Register | 0x0 |

### 17.4.1.1 TVE Video Format Control Register (Offset = 0x00)

**Table 17-2. TVE Video Format Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| 7 | Nonitl_263 | R/W | This bit is only valid when non-interlaced (Bit 5) is set to '1.'<br>0: 262 or 312 lines per field for NTSC or PAL non-interlaced mode<br>1: 263 or 313 lines per field for NTSC or PAL non-interlaced mode |
| 6 | subc | R/W | subcarrier phase work mode<br>0: Reset the subcarrier phase to zero at the beginning of each field sequence.<br>1: Do not reset the subcarrier phase at the beginning of each field sequence. |
| 5 | Non-interlaced | R/W | 0: Interlaced mode<br>1: Non-interlaced mode |
| 4 | - | - | Reserved |
| 3 | vin_range | R/W | The range of the video pixel input value<br>0: 16 to 235 for Y, 16 to 240 for Cr and Cb.<br>1: 1 to 254 for Y, Cr, and Cb |
| 2 | Black0IRE | R/W | Black level select<br>0: Black level is 7.5 IRE above blanking level<br>1: Black level is 0 IRE (Black level is the same as blanking level) |
| [1:0] | tvf[1:0] | R/W | TV standard format select (Should select correct **Black0IRE**)<br>00: NTSC-M,J (NTSC-M: Black level = 7.5 IRE, NTSC-J: Black level = 0 IRE)<br>01: PAL-M (Black level = 7.5 IRE)<br>10: PAL-B, D, G, H, I, and N (PAL-B,D,G,H,I: Black level = 0 IRE, PAL-N: Black level = 7.5 IRE)<br>11: PAL-$N_C$ (Black level = 0 IRE) |

### 17.4.1.2 TVE Video Input Control Register (Offset = 0x04)

**Table 17-3. TVE Video Input Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:7] | - | - | Reserved |
| [6:5] | vin_d[1:0] | R/W | Pixel data delay<br>In the video input BT601 slave mode (vinf[1:0] = 00), the first active pixel is sampled at the 245$^{th}$ or 265$^{th}$ clki rising edge for NTSC/PAL-M (525-line) or PAL (626-line) system respectively. vin_d is used to add delay to the duration between hsyncn_i and the first active pixel data. clki is 27 MHz.<br>00: Normal (no delay)<br>01: +1 clki cycle delay<br>10: +2 clki cycles delay<br>11: +3 clki cycles delay |
| 4 | hs_p | R/W | Polarity selection of hsyncn_i or hsyncn_o<br>0: hsyncn_i or hsyncn_o is falling edge active.<br>1: hsyncn_i or hsyncn_o is rising edge active. |
| 3 | vs_p | R/W | Polarity of vsyncn_i (or vsyncn_o)/FIELD<br>0: Low level of FIELD indicates odd fields and high level indicates even fields. (field_mode = FIELD function)<br>vsyncn_i (or vsyncn_o) is falling edge active. (field_mode = VSYNC function)<br>1: High level of FIELD indicates odd fields, and low level indicates even fields. (field_mode = FIELD function)<br>vsyncn_i (or vsyncn_o) is rising edge active. (field_mode = VSYNC function) |
| [2:1] | vinf[1:0] | R/W | Video input mode selection<br>00: BT601 slave mode: hsyncn_i + vsyncn_i (VSYNC/FIELD) based synchronization.<br>01: BT656 mode: ITU-R BT.656 format. In this mode, FTTVE100_S drives hsyncn_o and vsyncn_o (VSYNC/FIELD) output.<br>10: BT601 master mode: hsyncn_o + vsyncn_o (VSYNC/FIELD) based synchronization.<br>11: Internal 100% color bar pattern ("field_mode" should be set to '0'). |
| 0 | Field-mode | R/W | Select function of vsyncn_i or vsyncn_o signal<br>0: VSYNC function.<br>1: FIELD function. |

### 17.4.1.3 TVE Video Output Control Register (Offset = 0x08)

**Table 17-4. TVE Video Output Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:7] | - | - | Reserved |
| 6 | vof | R/W | Output format<br>0: CVBS and S-video output simultaneously<br>1: Component YCbCr output |
| 5 | blue | R/W | Enable blue color output. It is used to display blue screen while there is no video input signal.<br>0: Normal video output<br>1: Blue output in all output modes |
| 4 | black | R/W | Enable black output<br>0: Normal video output<br>1: Black output in all output modes |
| 3 | no_c | R/W | Disable color signal<br>0: Normal video output<br>1: Color signal is disabled on CVBS output. S-video or component output is normal. |
| [2:0] | y_d[2:0] | R/W | Luma relative to chroma timing on the CVBS (composite) output clki is 27 MHz.<br>000: Normal output (no delay)<br>001: +2 clki cycles delay<br>010: +4 clki cycles delay<br>011: +6 clki cycles delay<br>100: reserved<br>101: +2 clki cycles precedence<br>110: +4 clki cycles precedence<br>111: +6 clki cycles precedence |

### 17.4.1.4 TVE Power-Down Control Register (Offset = 0x0C)

**Table 17-5. TVE Power-down Control Registers**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:5] | - | - | Reserved |
| 4 | clki_p | R/W | Setting this bit to logic ‘1’ will reverse the phase of the clock input, clki. It is useful if the setup or hold time between clki and video input signals, including vsync, hsync, and pixel_data, is not enough.<br>0: The phase of internal clock is same as clki.<br>1: The phase of internal clock is the inverse of clki. |
| 3 | DAC_clk_p | R/W | Setting this bit to logic ‘1’ will reverse the phase of the clock output, clk_DAC. It is useful if the setup or hold time between clk_DAC and video output signals is insufficient.<br>0: The phase of clk_DAC is the same as the internal clock.<br>1: The phase of clk_DAC is the inverse of the internal clock. |
| 2 | DACv_pd | R/W | Power-down the CVBS output DAC<br>0: Normal operation<br>1: Power down the CVBS output DAC |
| 1 | DACy_pd | R/W | Power-down the Y output DAC<br>0: Normal operation<br>1: Power down the luminance (Y) output DAC |
| 0 | DACc_pd | R/W | Power-down the C output DAC<br>0: Normal operation<br>1: Power down the chrominance (C) output DAC |

### 17.4.1.5 TVE Software Reset Control Register (Offset = 0x10)

**Table 17-6. TVE Software Reset Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:1] | - | - | Reserved |
| 0 | sreset | W | Software reset<br>This bit will reset all registers to default reset values.<br>0: No reset<br>1: Software reset. Writing ‘1’ to this bit enables the software reset function and this bit will automatically reset to ‘0.’ It is not necessary to write ‘0’ to disable "sreset." |

# Chapter 18

# 3D-deinterlace De-noise Filter

This chapter contains the following sections:


- 18.1 General Description
- 18.2 Features
- 18.3 Block Diagram
- 18.4 Programming Model

## 18.1 General Description

The deinterlacing process is an important stage which converts ordinary TV interlaced sequences into progressive sequences for displaying on the progressive devices (e.g. Computers, LCD, Plasma Display, and Projection TV).

The de-noise process is critical in enhancing the scene quality and in reducing the encoded bit-rate in network. In the real time IP camera system, the de-noise function is very useful for eliminating the Gauss-distribution noise, so that the encoder can make lower effort to encode the source scene. On the other hand, the encoded files will be smaller and reduce the bit-rate transferred in network.
The 3D de-interlace and de-noise module adopts the pixel based measure to perform the intra and inter field interpolations, depending on the motion state which the pixel belongs to motion or stationary. Its goal is to generate a high quality progressive frame.

## 18.2 Features

- Built-in denoise filter to eliminate the high frequency noise of field difference
- Supports macroblock based picture with maximum resolution up to 720 x 480i
- Supports four-field motion detection (Forward forward/forward/current/next)
- Supports format of macroblock planar Y only, for YCbCr 444/422/420/400 source
- Supports format of macroblock semi-planar YCbCr, for YCbCr 422/420 source
- Supports format of raster packet YCbCr422, for packet YCbCr 422 source

## 18.3 Block Diagram

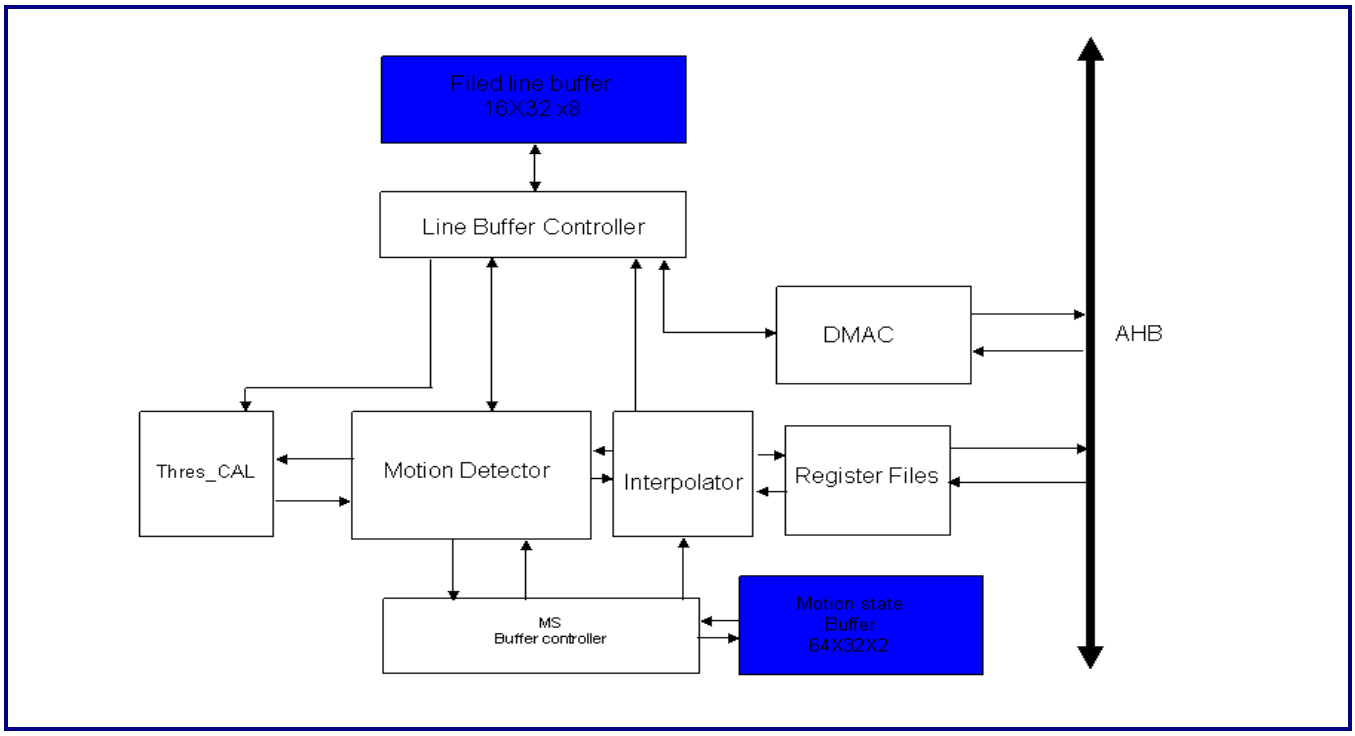



**Figure 18-1. Block Diagram**

## 18.4 Programming Model

### 18.4.1 Control Register Summary

**Table 18-1. Control Register Summary**

| Offset | Type | Width | Reset Value | Name | Config. | Description |
|---|---|---|---|---|---|---|
| 0x0000 | R/W | 32 | 0x0000_0000 | FFYA | None | Forward luma/Y start address (Top Frame) |
| 0x0004 | R/W | 32 | 0x0000_0000 | FWYA | None | Forward luma/Y start address (Bottom Field) |
| 0x0008 | R/W | 32 | 0x0000_0000 | CTYA | None | Current luma/Y start address (Top Frame) |
| 0x000C | R/W | 32 | 0x0000_0000 | NXYA | None | Next luma/Y start address (Bottom Field) |
| 0x0010 | R/W | 32 | 0x0000_0000 | CTCA | None | Current Chroma/CbCr start address (Top Frame) |
| 0x0014 | R/W | 32 | 0x0000_0000 | NXCA | None | Next Chroma/CbCr start address (Bottom Field) |
| 0x0018 | R/W | 32 | 0x0000_0000 | FFH | None | Frame/Field height |
| 0x001C | R/W | 32 | 0x0000_0000 | FW | None | Frame width |

| Offset | Type | Width | Reset Value | Name | Config. | Description |
|---|---|---|---|---|---|---|
| 0x0020 | R/W | 32 | 0x0000_0000 | COMD | None | Command register |
| 0x0024 | R/W | 32 | 0x0000_0000 | DSTY | None | Destination luma/Y start address |
| 0x0028 | R/W | 32 | 0x0000_0000 | DSTC | None | Destination Chroma/CbCr start address |
| 0x002C | R/W | 32 | 0x0000_0000 | THMDN | None | Thresholds of Denoise Filter/Motion Detection |
| 0x0030 | R/W | 32 | 0x0000_0000 | LMASK | None | Motion Detection MS Masks for low 32 pixels |
| 0x0034 | R/W | 32 | 0x0000_0000 | HMASK | None | Motion Detection MS Masks for high 32 pixels |
| 0x0038 | R/W | 32 | 0x2009_0120 | DATEID | None | Date/Version ID Information and RESET |
| 0x0040 | R | 32 | 0xXXXX_XXXX | STATUS | None | Internal Control status |
| 0x0050 | R/W | 32 | 0x0000_0000 | INTSTS | None | Interrupt status |
| 0x0100 | R/W | 32 | 0x0010_2008 | DENCFG | None | De-noise configure register |
| 0x0110 | R/W | 32 | 0x0820_8208 | TYVR0 | None | De-noise top luminance variances register0 |
| 0x0114 | R/W | 32 | 0x0820_8208 | TYVR1 | None | De-noise top luminance variances register1 |
| 0x0118 | R/W | 32 | 0x0820_8208 | TYVR2 | None | De-noise top luminance variances register2 |
| 0x011C | R/W | 32 | 0x0000_8208 | TYVR3 | None | De-noise top luminance variances register3 |
| 0x0120 | R/W | 32 | 0x0820_8208 | TCBVR0 | None | De-noise top chrominance (Cb) variances register0 |
| 0x0124 | R/W | 32 | 0x0820_8208 | TCBVR1 | None | De-noise top chrominance (Cb) variances register1 |
| 0x0128 | R/W | 32 | 0x0820_8208 | TCBVR2 | None | De-noise top chrominance (Cb) variances register2 |
| 0x012C | R/W | 32 | 0x0000_8208 | TCBVR3 | None | De-noise top chrominance (Cb) variances register3 |
| 0x0130 | R/W | 32 | 0x0820_8208 | TCRVR0 | None | De-noise top chrominance (Cr) variances register0 |
| 0x0134 | R/W | 32 | 0x0820_8208 | TCRVR1 | None | De-noise top chrominance (Cr) variances register1 |
| 0x0138 | R/W | 32 | 0x0820_8208 | TCRVR2 | None | De-noise top chrominance (Cr) variances register2 |
| 0x013C | R/W | 32 | 0x0000_8208 | TCRVR3 | None | De-noise top chrominance (Cr) variances register3 |
| 0x0140 | R/W | 32 | 0x0820_8208 | BYVR0 | None | De-noise bottom luminance variances register0 |
| 0x0144 | R/W | 32 | 0x0820_8208 | BYVR1 | None | De-noise bottom luminance variances register1 |
| 0x0148 | R/W | 32 | 0x0820_8208 | BYVR2 | None | De-noise bottom luminance variances register2 |
| 0x014C | R/W | 32 | 0x0000_8208 | BYVR3 | None | De-noise bottom luminance variances register3 |
| 0x0140 | R/W | 32 | 0x0820_8208 | BCBVR0 | None | De-noise bottom chrominance (Cb) variances register0 |
| 0x0144 | R/W | 32 | 0x0820_8208 | BCBVR1 | None | De-noise bottom chrominance (Cb) variances register1 |
| 0x0148 | R/W | 32 | 0x0820_8208 | BCBVR2 | None | De-noise bottom chrominance (Cb) variances register2 |
| 0x014C | R/W | 32 | 0x0000_8208 | BCBVR3 | None | De-noise bottom chrominance (Cb) variances register3 |
| 0x0150 | R/W | 32 | 0x0820_8208 | BCRVR0 | None | De-noise bottom chrominance (Cr) variances register0 |
| 0x0154 | R/W | 32 | 0x0820_8208 | BCRVR1 | None | De-noise bottom chrominance (Cr) variances register1 |
| 0x0158 | R/W | 32 | 0x0820_8208 | BCRVR2 | None | De-noise bottom chrominance (Cr) variances register2 |
| 0x015C | R/W | 32 | 0x0000_8208 | BCRVR3 | None | De-noise bottom chrominance (Cr) variances register3 |

### 18.4.2 Register Description

The following sections describe the detailed information of the FTDI210 registers.

#### 18.4.2.1 Forward Luma/Y Start address Register (Offset = 0x0000)

**Table 18-2. Forward Luma/Y Start address Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | FFY Address | R/W | Forward Luminance/Y Start Address (Macroblock Planar YCbCr Mode)<br>This 32-bit byte address specifies the luminance/Y start address of the forward picture (Top frame).<br>Forward Packet YCbCr Start Address (Raster Packet YCbCr 422 Mode)<br>This 32-bit byte address specifies the luminance/Y start address of the forward picture (Top frame). |

#### 18.4.2.2 Forward Luma/Y Start Address Register (Offset = 0x0004)

**Table 18-3. Forward Luma/Y Start Address Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | FWY Address | R/W | Forward Luminance/Y Start Address (Macroblock Planar YCbCr Mode)<br>This 32-bit byte address specifies the luminance/Y start address of the forward picture (Bottom field).<br>Forward Packet YCbCr Start Address (Raster Packet YCbCr 422 Mode)<br>This 32-bit byte address specifies the packet YCbCr 422 start address of the forward picture (Bottom field). |

#### 18.4.2.3 Current Luma/Y Start Address Register (Offset = 0x0008)

**Table 18-4. Current Luma/Y Start Address Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | CTY Address | R/W | Current Luminance/Y Start Address (Macroblock Planar YCbCr Mode)<br>This 32-bit byte address specifies the luminance/Y start address of the forward picture (Bottom field).<br>Current Packet YCbCr Start Address (Raster Packet YCbCr 422 Mode)<br>This 32-bit byte address specifies the packet YCbCr 422 start address of the current picture (Top frame). |

### 18.4.2.4 Next Luma/Y Start Address Register (Offset = 0x000C)

**Table 18-5. Next Luma/Y Start Address Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:0] | NXY Address | R/W | Next Luminance/Y Start Address (Macroblock Planar YCbCr)<br>This 32-bit byte address specifies the luminance/Y start address of the next picture (Bottom field).<br>Next Packet YCbCr Start Address (Raster Packet YCbCr 422)<br>This 32-bit byte address specifies the packet YCbCr 422 start address of the next picture (Bottom field). |

### 18.4.2.5 Current Chroma/CbCr Start Address Register (Offset = 0x0010)

**Table 18-6. Current Chroma/CbCr Start Address Register**

| Bit | Name | R/W | Description (Macroblock Planar YCbCr) |
|---|---|---|---|
| [31:0] | CTC Address | R/W | Current Chrominance/CbCr Start Address<br>This 32-bit byte address specifies the Chroma/CbCr start address of the current picture (Top frame). |

### 18.4.2.6 Next Chroma/CbCr Start Address Register (Offset = 0x0014)

**Table 18-7. Next Chroma/CbCr Start Address Register**

| Bit | Name | R/W | Description (Macroblock Planar YCbCr) |
|---|---|---|---|
| [31:0] | NXC Address | R/W | Next Chrominance/CbCr Start Address<br>This 32-bit byte address specifies the Chroma/CbCr start address of the next picture (Bottom field). |

### 18.4.2.7 Frame/Field Height Register (Offset = 0x0018)

**Table 18-8. Frame/Field Height Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:0] | Frame/Field Height | R/W | This 10-bit value specifies the frame/field height, and it should be a multiple of 16.<br>When command register's bit 4 (Source) is<br>0: This register should be input with field height.<br>1: This register should be input with frame height. |

### 18.4.2.8 Frame Width Register (Offset = 0x001C)

**Table 18-9. Frame Width Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:0] | Frame Width | R/W | This 10-bit value specifies the picture width, and it should be a multiple of 16. |

### 18.4.2.9 Command Register (Offset = 0x0020)

**Table 18-10. Command Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| 16 | mdt_adj | R/W | 0 = Auto adjustment in the motion detection threshold level<br>1 = Manual adjustment in the motion detection threshold level |
| 8 | denoise_only | R/W | 0 = Both the de-noise and de-interlace function are enable simultaneously<br>1 = The de-noise function is enable but the de-interlace function is off. |
| 7 | bypass | R/W | 0 = Normal deinterlacing process for Y/Cb/Cr components.<br>1 = Keep Y component only, that is, bypass Cb/Cr during the deinterlacing process. |
| 6 | mask_test | R/W | 0 = Disable the Motion/Static State mask specified by the LMASK/HMASK registers.<br>1 = Enable the Motion/Static State mask specified by the LMASK/HMASK registers. |
| 5 | target | R/W | 0 = Reconstruction or destination addresses of Y/Cb/Cr is hardwired with Capture.<br>1 = Reconstruction or destination addresses of Y/Cb/Cr is specified by the CPU (S/W). |
| 4 | source | R/W | 0 = Input frame/field or source addresses of Y/Cb/Cr is hardwired with Capture.<br>1 = Input frame/field or source addresses of Y/Cb/Cr is specified by the CPU (S/W). |
| 3 | swap_cbcr | R/W | 0 = Keep output pixel Cb/Cr position at the raster packet YCbCr 422 format.<br>1 = Swap output pixel Cb/Cr position at the raster packet YCbCr 422 format. |
| [2:1] | pk422/pl422/pl420# | R/W | 0 = Source picture format is macroblock planar YCbCr 420.<br>1 = Source picture format is macroblock planar YCbCr 422.<br>2 = Source picture format is raster packet YCbCr 422.<br>3= Source picture format is raster packet YCbCr 422. |
| 0 | cpu_fire | R/W | 0 = No action<br>1 = CPU write one to trigger di210 |

### 18.4.2.10 Destination Luma/Y Start Address Register (Offset = 0x0024)

**Table 18-11. Command Register**

| Bit | Name | R/W | Description (Macroblock Planar YCbCr) |
|---|---|---|---|
| [31:0] | DSTY Address | R/W | In Macroblock Planar YCbCr<br>Destination Luminance/Y Start Address<br>This 32-bit byte address specifies the luminance/Y start address of the current picture (Top frame).<br><br>In Raster Packet YCbCr 422<br>Destination Packet YCbCr Start Address<br>This 32-bit byte address specifies the packet YCbCr 422 start address of the current picture (Top frame). |

### 18.4.2.11 Destination Chroma/CbCr Start Address Register (Offset = 0x0028)

**Table 18-12. Destination Chroma/CbCr Start Address Register**

| Bit | Name | R/W | Description (Macroblock Planar YCbCr) |
|---|---|---|---|
| [31:0] | DSTC Address | R/W | Destination Chrominance/CbCr Start Address<br>This 32-bit byte address specifies the chrominace/CbCr start address of the current picture (Top frame). |

### 18.4.2.12 Destination Chroma/CbCr Start Address Register (Offset = 0x002C)

**Table 18-13. Destination Chroma/CbCr Start Address Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [23:16] | MD_TH | R/W | Motion Detection Threshold<br>This 8-bit value specifies the manual adjustment threshold level during the MD Cycle if Command Register bit-16 mdt_adj = 1. |
| [3:0] | MB_TH | R/W | Motion Block Threshold<br>This 4-bit value specifies the threshold level of the macroblock motion judgment for the motion filter. |

### 18.4.2.13 Motion Detection MS Mask Register for Low 32 Pixels (Offset = 0x0030)

**Table 18-14. Motion Detection MS Mask Register for Low 32 Pixels**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:0] | Low Mask | R/W | Motion/Static State Low Mask<br>It specifies the motion/static state mask of low 32 pixels when mask_test = 1. |

### 18.4.2.14 Motion Detection MS Mask Register for High 32 Pixels (Offset = 0x0034)

**Table 18-15. Motion Detection MS Mask Register for High 32 Pixels**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:0] | High Mask | R/W | Motion/Static State High Mask<br>It specifies the motion/static state mask of high 32 pixels when mask_test = 1. |

### 18.4.2.15 Date and ID Information Register (Offset = 0x0038)

**Table 18-16. Date and ID Information Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:16] | DATE | R | It specifies the data information. |
| 15:8 | VERSION ID | R | It specifies the Version ID information. |

### 18.4.2.16 Internal Control Status Register (Offset = 0x003C)

**Table 18-17. Internal Control Status Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:0] | STATUS | R | Internal status used for debugging. |

### 18.4.2.17 Interrupt Status Register (Offset = 0x0050)

**Table 18-18. Internal Control Status Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| 0 | INTS | R/W | 0 = No interrupt occurs<br>1 = Interrupt occurs, the host writes ‘1’ to clear INTS |

### 18.4.2.18 De-noise Conifigure Register (Offset = 0x0100)

**Table 18-19. De-noise Conifigure Register**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [22:20] | Mtau Selection | R/W | 0 = If (abs(delta) ≥ 2var(k)), tau = 1<br>1 = If (abs(delta) ≥ 2var(k)), tau = 2<br>2 = If (abs(delta) ≥ 2var(k)), tau = 4<br>3 = If (abs(delta) ≥ 2var(k)), tau = 8<br>4 = If (abs(delta) ≥ 2var(k)), tau = 16<br>5 = If (abs(delta) ≥ 2var(k)), tau = 32<br>Others= If (abs(delta) ≥ 2var(k)), tau = 2 |
| [17:16] | Gama Selection | R/W | 0 = Gama = 1.5<br>1 = Gama = 2<br>2 = Gama = 2.5<br>3 = Gama = 4 |
| [14:8] | Learn Rate | R/W | Learn rate value decide the variance changing rate when delta is greater than clip value-1. |
| [4:0] | Clip value | R/W | Clip value is the threshold to decide updating variance or not |

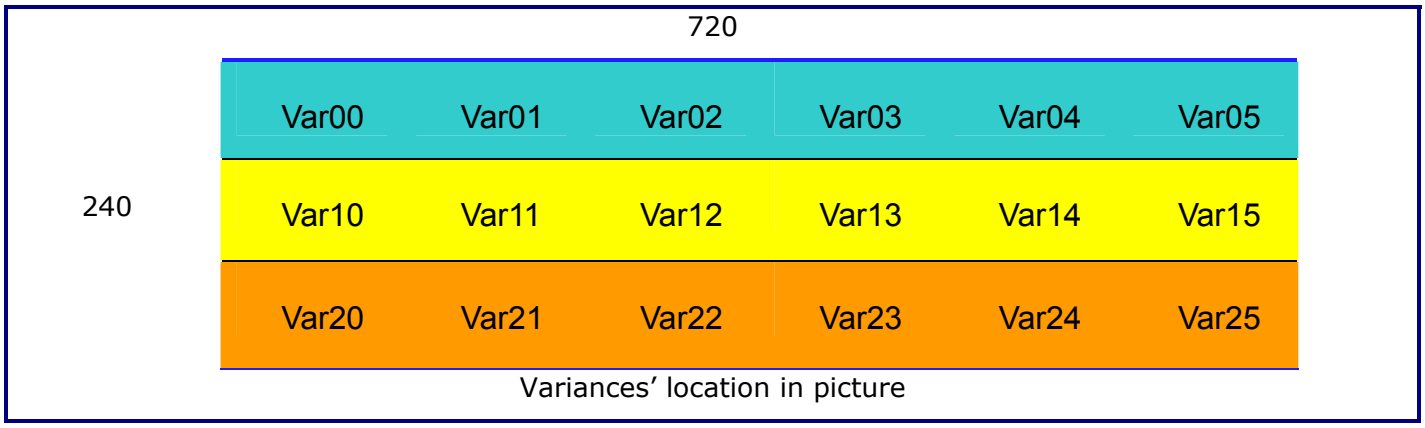
Reset Value：0x0010_2008

### 18.4.2.19 De-noise Top Luma Variance Register0 (Offset = 0x0110)

**Table 18-20. De-noise Top Luma Variance Register0**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:30] | | - | Reserved |
| [29:24] | TYvar04 | R/W | Top field luminance variance variable located in row 0 and column 4 |
| [23:18] | TYvar03 | R/W | Top field luminance variance variable located in row 0 and column 3 |
| [17:12] | TYvar02 | R/W | Top field luminance variance variable located in row 0 and column 2 |
| [11:6] | TYvar01 | R/W | Top field luminance variance variable located in row 0 and column 1 |
| [5:0] | TYvar00 | R/W | Top field luminance variance00 located in row 0 and column 0 (Variance width = 120, height = 80) |

Reset Value：0x0820_8208




### 18.4.2.20 De-noise Top Luma Variance Register1 (Offset = 0x0114)

**Table 18-21. De-noise Top Luma Variance Register1**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:24] | TYvar13 | R/W | Top field luminance variance variable located in row 1 and column 3 |
| [23:18] | TYvar12 | R/W | Top field luminance variance variable located in row 1 and column 2 |
| [17:12] | TYvar11 | R/W | Top field luminance variance variable located in row 1 and column 1 |
| [11:6] | TYvar10 | R/W | Top field luminance variance variable located in row 1 and column 0 |
| [5:0] | TYvar05 | R/W | Top field luminance variance located in row 0 and column 5 |

Reset Value: 0x0820_8208

### 18.4.2.21 De-noise Top Luma Variance Register2 (Offset = 0x0118)

**Table 18-22. De-noise Top Luma Variance Register2**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:24] | TYvar22 | R/W | Top field luminance variance variable located in row 2 and column 2 |
| [23:18] | TYvar21 | R/W | Top field luminance variance variable located in row 2 and column 1 |
| [17:12] | TYvar20 | R/W | Top field luminance variance variable located in row 2 and column 0 |
| [11:6] | TYvar15 | R/W | Top field luminance variance variable located in row 1 and column 5 |
| [5:0] | TYvar14 | R/W | Top field luminance variance located in row 1 and column 4 |

Reset Value: 0x0820_8208

### 18.4.2.22 De-noise Top Luma Variance Register3 (Offset = 0x011C)

**Table 18-23. De-noise Top Luma Variance Register3**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:18] | - | - | Reserved |
| [17:12] | TYvar25 | R/W | Top field luminance variance variable located in row 2 and column 5 |
| [11:6] | TYvar24 | R/W | Top field luminance variance variable located in row 2 and column 4 |
| [5:0] | TYvar23 | R/W | Top field luminance variance located in row 2 and column 3 |

Reset Value: 0x0000_8208

### 18.4.2.23 De-noise Top Chroma (Cb) Variance Register0 (Offset = 0x0120)

**Table 18-24. De-noise Top Chroma(Cb) Variance Register0**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31-30] | - | - | Reserved |
| [29:24] | TCbvar04 | R/W | Top field chrominance (Cb) variance variable located in row 0 and column 4 |
| [23:18] | TCbvar03 | R/W | Top field chrominance (Cb) variance variable located in row 0 and column 3 |
| [17:12] | TCbvar02 | R/W | Top field chrominance (Cb) variance variable located in row 0 and column 2 |
| [11:6] | TCbvar01 | R/W | Top field chrominance (Cb) variance variable located in row 0 and column 1 |
| [5:0] | TCbvar00 | R/W | Top field chrominance (Cb) variance located in row 0 and column 0 (Variance width = 120, height = 80) |

Reset Value: 0x0820_8208

### 18.4.2.24 De-noise Top Chroma (Cb) Variance Register1 (Offset = 0x0124)

**Table 18-25. De-noise Top Chroma (Cb) Variance Register1**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:24] | TCbvar13 | R/W | Top field chrominance (Cb) variance variable located in row 1 and column 3 |
| [23:18] | TCbvar12 | R/W | Top field chrominance (Cb) variance variable located in row 1 and column 2 |
| [17:12] | TCbvar11 | R/W | Top field chrominance (Cb) variance variable located in row 1 and column 1 |
| [11:6] | TCbvar10 | R/W | Top field chrominance (Cb) variance variable located in row 1 and column 0 |
| [5:0] | TCbvar05 | R/W | Top field chrominance (Cb) variance located in row 0 and column 5 |

Reset Value: 0x0820_8208

### 18.4.2.25 De-noise Top Chroma (Cb) Variance Register2 (Offset = 0x0128)

**Table 18-26. De-noise Top Chroma (Cb) Variance Register2**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:24] | TCbvar22 | R/W | Top field chrominance(Cb) variance variable located in row 2 and column 2 |
| [23:18] | TCbvar21 | R/W | Top field chrominance (Cb) variance variable located in row 2 and column 1 |
| [17:12] | TCbvar20 | R/W | Top field chrominance (Cb) variance variable located in row 2 and column 1 |
| [11:6] | TCbvar15 | R/W | Top field chrominance (Cb) variance variable located in row 1 and column 5 |
| [5:0] | TCbvar14 | R/W | Top field chrominance (Cb) variance located in row 1 and column 4 |

Reset Value: 0x0820_8208

### 18.4.2.26 De-noise Top Chroma(Cb) Variance Register3 (Offset = 0x012C)

**Table 18-27. De-noise Top Chroma (Cb) Variance Register3**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:18] | - | - | Reserved |
| [17:12] | TCbvar20 | R/W | Top field chrominance (Cb) variance variable located in row 2 and column 1 |
| [11:6] | TCbvar24 | R/W | Top field chrominance (Cb) variance variable located in row 2 and column 4 |
| [5:0] | TCbvar23 | R/W | Top field chrominance (Cb) variance located in row 2 and column 3 |

Reset Value: 0x0000_8208

### 18.4.2.27 De-noise Top Chroma (Cr) Variance Register0 (Offset = 0x0130)

**Table 18-28. De-noise Top Chroma (Cr) Variance Register0**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:18] | - | - | Reserved |
| [29:24] | TCrvar04 | R/W | Top field chrominance (Cr) variance variable located in row 0 and column 4 |
| [23:18] | TCrvar03 | R/W | Top field chrominance (Cr) variance variable located in row 0 and column 3 |
| [17:12] | TCrvar02 | R/W | Top field chrominance( Cr) variance variable located in row 0 and column 2 |
| [11:6] | TCrvar01 | R/W | Top field chrominance (Cr) variance variable located in row 0 and column 1 |

| Bit | Name | R/W | Description |
|---|---|---|---|
| [5:0] | TCrvar00 | R/W | Top field chrominance (Cr) variance located in row 0 and column 0 (Variance width = 120, height = 80) |

Reset Value: 0x0820_8208

### 18.4.2.28 De-noise Top Chroma (Cr) Variance Register1 (Offset = 0x0134)

**Table 18-29. De-noise Top Chroma (Cr) Variance Register1**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:24] | TCrvar13 | R/W | Top field chrominance (Cr) variance variable located in row 1 and column 3 |
| [23:18] | TCrvar12 | R/W | Top field chrominance (Cr) variance variable located in row 1 and column 2 |
| [17:12] | TCrvar11 | R/W | Top field chrominance (Cr) variance variable located in row 1 and column 1 |
| [11:6] | TCrvar10 | R/W | Top field chrominance (Cr) variance variable located in row 1 and column 0 |
| [5:0] | TCrvar05 | R/W | Top field chrominance (Cr) variance located in row 0 and column 5 (Variance width = 120, height = 80) |

Reset Value: 0x0820_8208

### 18.4.2.29 De-noise Top Chroma (Cr) Variance Register2 (Offset = 0x0138)

**Table 18-30. De-noise Top Chroma (Cr) Variance Register2**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:30] | - | - | Reserved |
| [29:24] | TCrvar22 | R/W | Top field chrominance (Cr) variance variable located in row 2 and column 2 |
| [23:18] | TCrvar21 | R/W | Top field chrominance (Cr) variance variable located in row 2 and column 1 |
| [17:12] | TCrvar20 | R/W | Top field chrominance (Cr) variance variable located in row 2 and column 0 |
| [11:6] | TCrvar15 | R/W | Top field chrominance (Cr) variance variable located in row 1 and column 5 |
| [5:0] | TCrvar14 | R/W | Top field chrominance (Cr) variance located in row 1 and column 4 (Variance width = 120, height = 80) |

Reset Value: 0x0820_8208

### 18.4.2.30 De-noise Top Chroma (Cr) Variance Register3 (Offset = 0x013C)

**Table 18-31. De-noise Top Chroma (Cr) Variance Register3**

| Bit | Name | R/W | Description |
|---|---|---|---|
| [31:18] | - | - | Reserved |
| [17:12] | TCrvar25 | R/W | Top field chrominance(Cr) variance variable located in row 2 and column 5 |
| [11:6] | TCrvar24 | R/W | Top field chrominance(Cr) variance variable located in row 2 and column 4 |
| [5:0] | TCrvar23 | R/W | Top field chrominance(Cr) variance located in row 2 and column 3 (Variance width = 120, height = 80) |

Reset Value: 0x0000_8208

# Chapter 19

# Video Capture

This chapter contains the following sections:


- 19.1 General Description
- 19.2 Features
- 19.3 Block Diagram
- 19.4 Memory Map/Register Definition
- 19.5 Function Description

## 19.1 General Description

FTVCAP210 is used to capture the video data from various video interfaces and output the data to AMBA AHB. It provides the de-interlace function for reducing the video artifact of the interlace video. With the size-down function, users can change the size of an image to the necessary resolution and that can be done individually for two paths. The Color OSD function helps user paste any characters of the captured video. The window clipping function clips the interested region of an image before or after size down. The host processor interface is compliant with AMBA AHB 2.0, which can serve as an IP core for integrating into the AMBA system.

## 19.2 Features

- Maximum capture resolution of up to 1920 x 1080
- Supports ITU-R BT. 656 8-bit input interface
- Supports ITU-R BT. 1120 16-bit input interface
- Supports 8-bit/16-bit YCbCr 4:2:2 with H/V reference control signal interface
- One video capture input with two data output paths
- Supports edge-based line average de-interlacer
- Includes embedded font RAM for color OSD at path 0 and path 1
- Supports up to 8 mask windows with transparency degree control
- Supports size-down function for individual image at two output paths
  - Path 0: Supports integer sizing down ratio
  - Path 1: Supports fraction sizing down ratio
- Supports frame-skip function at two output paths
- Supports even/odd field-sorting function at two output paths
- Supports VBI data extraction function
- Supports source and output target image crop function
- Supports YCbCr 4:2:2 output format for two output paths
- Supports DMA raster order output sequence and output frame flip function
- Includes capture clock up to 108 MHz and AHB clock up to 240 MHz for 90 nm process

## 19.3 Block Diagram

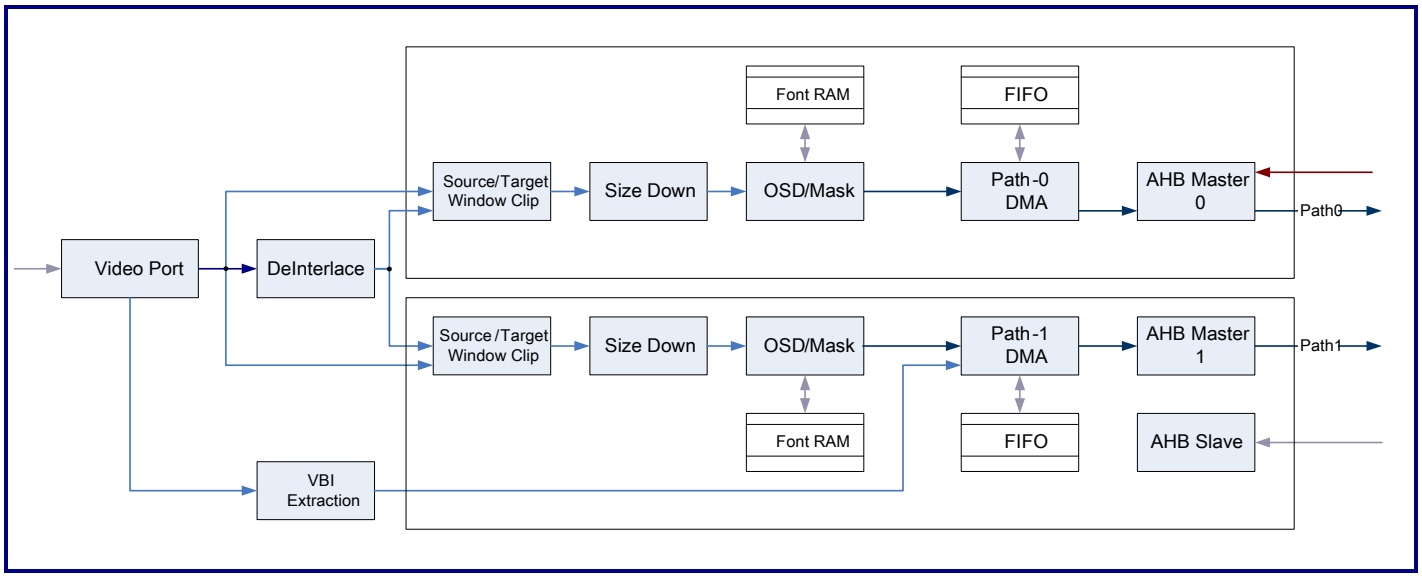



**Figure 19-1. Block Diagram**

## 19.4 Memory Map/Register Definition

### 19.4.1 Mapping of Usual Special-purpose Registers

**Table 19-1. Mapping of Usual Special-purpose Registers**

| Symbol | Offset | Access | Description |
|---|---|---|---|
| VCAPEN | 0x0000 | R/W | Basic function enable and trigger control |
| VCAPUPD | 0x0004 | R/W | Register update control |
| CAPCLK | 0x0008 | R/W | Capture clock setting |
| P0SIZE0 | 0x0010 | R/W | Output target image width and height on path 0 |
| P0SIZE1 | 0x0014 | R/W | Top-left corner of output crop window on path 0 |
| P0SIZE2 | 0x0018 | R/W | Output crop width and height on path 0 |
| P1SIZE0 | 0x0020 | R/W | Output target image width and height on path 1 |
| P1SIZE1 | 0x0024 | R/W | Top-left corner of output crop window on path 1 |
| P1SIZE2 | 0x0028 | R/W | Output crop width and height on path 1 |
| DFORMAT | 0x0038 | R/W | Data output format |
| GBATRB | 0x003C | R/W | Global attribute |
| PTATRB | 0x0040 | R/W | Path attribute |

| Symbol | Offset | Access | Description |
|---|---|---|---|
| SRCIF | 0x0044 | R/W | Source interface related control |
| SRCSIZE0 | 0x0050 | R/W | Source image width and height |
| P0SIZE3 | 0x0054 | R/W | Top-left corner of source active window on path 0 |
| P0SIZE4 | 0x0058 | R/W | Source active window width and height on path 0 |
| P0SIZE5 | 0x005C | R | Image size after sizing down on path 0 |
| SRCSIZE1 | 0x0060 | R/W | Horizontal and vertical pixel active point of source image |
| P1SIZE3 | 0x0064 | R/W | Top-left corner of source active window on path 1 |
| P1SIZE4 | 0x0068 | R/W | Source active window width and height on path 1 |
| P1SIZE5 | 0x006C | R | Image size after sizing down on path 1 |
| P1SHARPEN | 0x00D8 | R/W | Sharpen on path1 |
| P0MSK0COR | 0x0100 | R/W | Mask color of window 0 on path 0 |
| P0MSK0SIZE0 | 0x0104 | R/W | Mask window 0 width on path 0 |
| P0MSK0SIZE1 | 0x0108 | R/W | Mask window 0 height on path 0 |
| P0MSK1COR | 0x010C | R/W | Mask color of window 1 on path 0 |
| P0MSK1SIZE0 | 0x0110 | R/W | Mask window 1 width on path 0 |
| P0MSK1SIZE1 | 0x0114 | R/W | Mask window 1 height on path 0 |
| P0MSK2COR | 0x0118 | R/W | Mask color of window 2 on path 0 |
| P0MSK2SIZE0 | 0x011C | R/W | Mask window 2 width on path 0 |
| P0MSK2SIZE1 | 0x0120 | R/W | Mask window 2 height on path 0 |
| P0MSK3COR | 0x0124 | R/W | Mask color of window 3 on path 0 |
| P0MSK3SIZE0 | 0x0128 | R/W | Mask window 3 width on path 0 |
| P0MSK3SIZE1 | 0x012C | R/W | Mask window 3 height on path 0 |
| P0MSK4COR | 0x0130 | R/W | Mask color of window 4 on path 0 |
| P0MSK4SIZE0 | 0x0134 | R/W | Mask window 4 width on path 0 |
| P0MSK4SIZE1 | 0x0138 | R/W | Mask window 4 height on path 0 |
| P0MSK5COR | 0x013C | R/W | Mask color of window 5 on path 0 |
| P0MSK5SIZE0 | 0x0140 | R/W | Mask window 5 width on path 0 |
| P0MSK5SIZE1 | 0x0144 | R/W | Mask window 5 height on path 0 |
| P0MSK6COR | 0x0148 | R/W | Mask color of window 6 on path 0 |
| P0MSK6SIZE0 | 0x014C | R/W | Mask window 6 width on path 0 |
| P0MSK6SIZE1 | 0x0150 | R/W | Mask window 6 height on path 0 |
| P0MSK7COR | 0x0154 | R/W | Mask color of window 7 on path 0 |
| P0MSK7SIZE0 | 0x0158 | R/W | Mask window 7 width on path 0 |
| P0MSK7SIZE1 | 0x015C | R/W | Mask window 7 height on path 0 |
| P1MSK0COR | 0x0160 | R/W | Mask color of window 0 on path 1 |
| P1MSK0SIZE0 | 0x0164 | R/W | Mask window 0 width on path 1 |

| Symbol | Offset | Access | Description |
|---|---|---|---|
| P1MSK0SIZE1 | 0x0168 | R/W | Mask window 0 height on path 1 |
| P1MSK1COR | 0x016C | R/W | Mask color of window 1 on path 1 |
| P1MSK1SIZE0 | 0x0170 | R/W | Mask window 1 width on path 1 |
| P1MSK1SIZE1 | 0x0174 | R/W | Mask window 1 height on path 1 |
| P1MSK2COR | 0x0178 | R/W | Mask color of window 2 on path 1 |
| P1MSK2SIZE0 | 0x017C | R/W | Mask window 2 width on path 1 |
| P1MSK2SIZE1 | 0x0180 | R/W | Mask window 2 height on path 1 |
| P1MSK3COR | 0x0184 | R/W | Mask color of window 3 on path 1 |
| P1MSK3SIZE0 | 0x0188 | R/W | Mask window 3 width on path 1 |
| P1MSK3SIZE1 | 0x018C | R/W | Mask window 3 height on path 1 |
| P1MSK4COR | 0x0190 | R/W | Mask color of window 4 on path 1 |
| P1MSK4SIZE0 | 0x0194 | R/W | Mask window 4 width on path 1 |
| P1MSK4SIZE1 | 0x0198 | R/W | Mask window 4 height on path 1 |
| P1MSK5COR | 0x019C | R/W | Mask color of window 5 on path 1 |
| P1MSK5SIZE0 | 0x01A0 | R/W | Mask window 5 width on path 1 |
| P1MSK5SIZE1 | 0x01A4 | R/W | Mask window 5 height on path 1 |
| P1MSK6COR | 0x01A8 | R/W | Mask color of window 6 on path 1 |
| P1MSK6SIZE0 | 0x01AC | R/W | Mask window 6 width on path 1 |
| P1MSK6SIZE1 | 0x01B0 | R/W | Mask window 6 height on path 1 |
| P1MSK7COR | 0x01B4 | R/W | Mask color of window 7 on path 1 |
| P1MSK7SIZE0 | 0x01B8 | R/W | Mask window 7 width on path 1 |
| P1MSK7SIZE1 | 0x01BC | R/W | Mask window 7 height on path 1 |
| DICTRL0 | 0x01E0 | R/W | 2D de-interlace control |
| DICTRL1 | 0x01E4 | R/W | 2D de-interlace control |
| MISC0 | 0x01F0 | R/W | Miscellaneous register. |
| P0DMA0 | 0x0200 | R/W | DMA setting on path 0 |
| P1DMA0 | 0x0204 | R/W | DMA setting on path 1 |
| P0DMA1 | 0x0218 | R/W | DMA setting on path 0 |
| P1DMA1 | 0x021C | R/W | DMA setting on path 1 |
| P0MDEST0 | 0x0220 | R/W | Destination frame buffer 0 Y start address for path 0 |
| P0MDEST1 | 0x0224 | R/W | Destination frame buffer 1 Y start address for path 0 |
| P0MDEST2 | 0x0230 | R/W | Destination frame buffer 0 Cb start address for path 0 |
| P0MDEST3 | 0x0234 | R/W | Destination frame buffer 1 Cb start address for path 0 |
| P0MDEST4 | 0x0240 | R/W | Destination frame buffer 0 Cr start address for path 0 |
| P0MDEST5 | 0x0244 | R/W | Destination frame buffer 1 Cr start address for path 0 |
| P1MDEST0 | 0x0250 | R/W | Destination frame buffer 0 Y start address for path 1 |

| Symbol | Offset | Access | Description |
|---|---|---|---|
| P1MDEST1 | 0x0254 | R/W | Destination frame buffer 1 Y start address for path 1 |
| P1MDEST2 | 0x0260 | R/W | Destination frame buffer 0 Cb start address for path 1 |
| P1MDEST3 | 0x0264 | R/W | Destination frame buffer 1 Cb start address for path 1 |
| P1MDEST4 | 0x0270 | R/W | Destination frame buffer 0 Cr start address for path 1 |
| P1MDEST5 | 0x0274 | R/W | Destination frame buffer 1 Cr start address for path 1 |
| VBICTRL0 | 0x0290 | R/W | VBI frame numbers 0 start address for path-1 DMA |
| VBICTRL1 | 0x0294 | R/W | VBI frame numbers 1 start address for path-1 DMA |
| VBICTRL2 | 0x02A0 | R/W | VBI stat line and pixel locations |
| VBICTRL3 | 0x02B0 | R/W | VBI size |
| P0OSDFONT | 0x0300 | R/W | OSD font RAM address and data port on path 0 |
| P0OSDDISP | 0x0304 | R/W | OSD display RAM address and data port on path 0 |
| P0OSDREAD | 0x0308 | R | OSD font RAM and display RAM data read port on path 0 |
| P0OSDEN | 0x030C | R/W | OSD window 0 ~ window 3 enable on path 0 |
| P0OSDPAT0 | 0x0310 | R/W | OSD palette color 0 on path 0 |
| P0OSDPAT1 | 0x0314 | R/W | OSD palette color 1 on path 0 |
| P0OSDPAT2 | 0x0318 | R/W | OSD palette color 2 on path 0 |
| P0OSDPAT3 | 0x031C | R/W | OSD palette color 3 on path 0 |
| P0OSDPAT4 | 0x0320 | R/W | OSD palette color 4 on path 0 |
| P0OSDPAT5 | 0x0324 | R/W | OSD palette color 5 on path 0 |
| P0OSDPAT6 | 0x0328 | R/W | OSD palette color 6 on path 0 |
| P0OSDCOR0 | 0x0330 | R/W | OSD font color of window 0 on path 0 |
| P0OSDWSZ0 | 0x0334 | R/W | OSD window 0 width/height on path 0 |
| P0OSDSSZ0 | 0x0338 | R/W | OSD window 0 start point on path 0 |
| P0OSDFSZ0 | 0x033C | R/W | OSD font size of window 0 on path 0 |
| P0OSDCOL1 | 0x0340 | R/W | OSD font color of window 1 on path 0 |
| P0OSDWSZ1 | 0x0344 | R/W | OSD window 1 width/height on path 0 |
| P0OSDSSZ1 | 0x0348 | R/W | OSD window 1 start point on path 0 |
| P0OSDFSZ1 | 0x034C | R/W | OSD font size of window 1 on path 0 |
| P0OSDCOR2 | 0x0350 | R/W | OSD font color of window 2 on path 0 |
| P0OSDWSZ2 | 0x0354 | R/W | OSD window 2 width/height on path 0 |
| P0OSDSSZ2 | 0x0358 | R/W | OSD window 2 start point on path 0 |
| P0OSDFSZ2 | 0x035C | R/W | OSD font size of window 2 on path 0 |
| P0OSDCOR3 | 0x0360 | R/W | OSD font color of window 3 on path 0 |
| P0OSDWSZ3 | 0x0364 | R/W | OSD window 3 width/height on path 0 |
| P0OSDSSZ3 | 0x0368 | R/W | OSD window 3 start point on path 0 |
| P0OSDFSZ3 | 0x036C | R/W | OSD font size of window 3 on path 0 |

| Symbol | Offset | Access | Description |
|---|---|---|---|
| P1OSDFONT | 0x0370 | R/W | OSD font RAM address and data port on path 1 |
| P1OSDDISP | 0x0374 | R/W | OSD display RAM address and data port on path 1 |
| P1OSDREAD | 0x0378 | R | OSD font RAM and display RAM data read port on path 1 |
| P1OSDEN | 0x037C | R/W | OSD window 0 ~ window 3 enable on path 1 |
| P1OSDPAT0 | 0x0380 | R/W | OSD palette color 0 on path 1 |
| P1OSDPAT1 | 0x0384 | R/W | OSD palette color 1 on path 1 |
| P1OSDPAT2 | 0x0388 | R/W | OSD palette color 2 on path 1 |
| P1OSDPAT3 | 0x038C | R/W | OSD palette color 3 on path 1 |
| P1OSDPAT4 | 0x0390 | R/W | OSD palette color 4 on path 1 |
| P1OSDPAT5 | 0x0394 | R/W | OSD palette color 5 on path 1 |
| P1OSDPAT6 | 0x0398 | R/W | OSD palette color 6 on path 1 |
| P1OSDCOR0 | 0x03A0 | R/W | OSD font color of window 0 on path 1 |
| P1OSDWSZ0 | 0x03A4 | R/W | OSD window 0 width/height on path 1 |
| P1OSDSSZ0 | 0x03A8 | R/W | OSD window 0 start point on path 1 |
| P1OSDFSZ0 | 0x03AC | R/W | OSD font size of window 0 on path 1 |
| P1OSDCOL1 | 0x03B0 | R/W | OSD font color of window 1 on path 1 |
| P1OSDWSZ1 | 0x03B4 | R/W | OSD window 1 width/height on path 1 |
| P1OSDSSZ1 | 0x03B8 | R/W | OSD window 1 start point on path 1 |
| P1OSDFSZ1 | 0x03BC | R/W | OSD font size of window 1 on path 1 |
| P1OSDCOR2 | 0x03C0 | R/W | OSD font color of window 2 on path 1 |
| P1OSDWSZ2 | 0x03C4 | R/W | OSD window 2 width/height on path 1 |
| P1OSDSSZ2 | 0x03C8 | R/W | OSD window 2 start point on path 1 |
| P1OSDFSZ2 | 0x03CC | R/W | OSD font size of window 2 on path 1 |
| P1OSDCOR3 | 0x03D0 | R/W | OSD font color of window 3 on path 1 |
| P1OSDWSZ3 | 0x03D4 | R/W | OSD window 3 width/height on path 1 |
| P1OSDSSZ3 | 0x03D8 | R/W | OSD window 3 start point on path 1 |
| P1OSDFSZ3 | 0x03DC | R/W | OSD font size of window 3 on path 1 |
| INTSTS | 0x03E0 | R/W | Status register |
| INTMASK | 0x03E4 | R/W | Interrupt mask register |

## 19.4.2 Special-purpose Registers and Functions

### 19.4.2.1 VCAPEN (Offset: 0x0000)

**Table 19-2. VCAPEN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | CAPTURE_EN | R/W | 0 | Image data capture control<br>0: Disable<br>1: Enable |
| 30 | - | - | - | Reserved |
| 29 | FM_STEP_CAP | R/W | 0 | Continuous or Step capture model control<br>0: Continuous frame capture mode<br>1: Single frame capture mode |
| 28 | ALBUM_EN | R/W | 0 | Photo album capture mode control (Sync. register)<br>0: Disable<br>1: Enable |
| [27:26] | - | - | - | Reserved |
| 25 | STEP_CAP_FIRE | R/W | 0 | Stepping-frame capture fire (Auto clear)<br>1: Fire one-step capture |
| 24 | PIMG_FIRE | R/W | 0 | Post-image processing fire (Auto clear)<br>1: Fire post-image processing |
| [23:22] | - | - | - | Reserved |
| [21:20] | LINKLIST_EN | R/W | 0 | Link List function control<br>00: Disable Link List function<br>01: Read Link List of Path 0 before capturing every frame<br>10: Read Link List of Path 1 before capturing every frame<br>11: Read Link List of both paths before capturing every frame |
| 19 | - | - | - | Reserved |
| [18:16] | DI_EN | R/W | 0 | De-interlace (DI) function control<br>0: Disable<br>1: Reserved<br>2: Enable |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 15 | DI_ELA_EN | R/W | 1 | DI ELA function control<br>0: Disable<br>DI duplicates the even or odd line.<br>1: Enable |
| [14:13] | - | - | - | Reserved |
| 12 | VP_EC_EN | R/W | 1 | Error correction control of the CCIR 656 SAV/EAV code<br>0: Disable<br>1: Enable |
| [11:10] | - | - | - | Reserved |
| 9 | US_EN | R/W | 0 | Cb/Cr Up-sampling function<br>0: Disable<br>1: Enable |
| 8 | DITHER_EN | R/W | 1 | Dither function control of the DT module<br>0: Disable<br>1: Enable |
| 7 | - | - | - | Reserved |
| 6 | SP_EN | R/W | 1 | Sharpness function control of Path 1 (Sync register)<br>0: Disable<br>1: Enable |
| [5:0] | - | - | - | Reserved |

### 19.4.2.2 VCAPUPD (Offset: 0x0004)

**Table 19-3. VCAPUPD Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| 24 | FORCE_LLI | R/W | 0 | Force LLI to read Link List (Auto clear)<br>0: No operation<br>1: Force LLI to read Link List |
| [23:17] | - | - | - | Reserved |
| 16 | FORCE_CU | R/W | 0 | Force CU to perform calculation (Auto clear)<br>0: CU performs calculation in the vertical blank interval.<br>1: Force CU to perform calculation |
| [15:13] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 12 | FORCE_P1_UPD | R/W | 0 | Force all “path 1” sync registers to perform update<br>(Auto Clear)<br>0: Do not update the sync. registers<br>1: All p1 sync. registers will be updated immediately |
| [11:9] | - | - | - | Reserved |
| 8 | FORCE_P0_UPD | R/W | 0 | Force all “path 0” sync registers to perform update<br>(Auto Clear)<br>0: Do not update the sync. registers<br>1: All p0 sync. registers will be updated immediately |
| [7:2] | - | - | - | Reserved |
| 1 | FORCE_UPDATE | R/W | 0 | Force all sync. registers, except for the p0/p1 sync. register, to perform update (Auto clear)<br>0: Do not update the sync. registers<br>1: Update all sync. registers immediately |

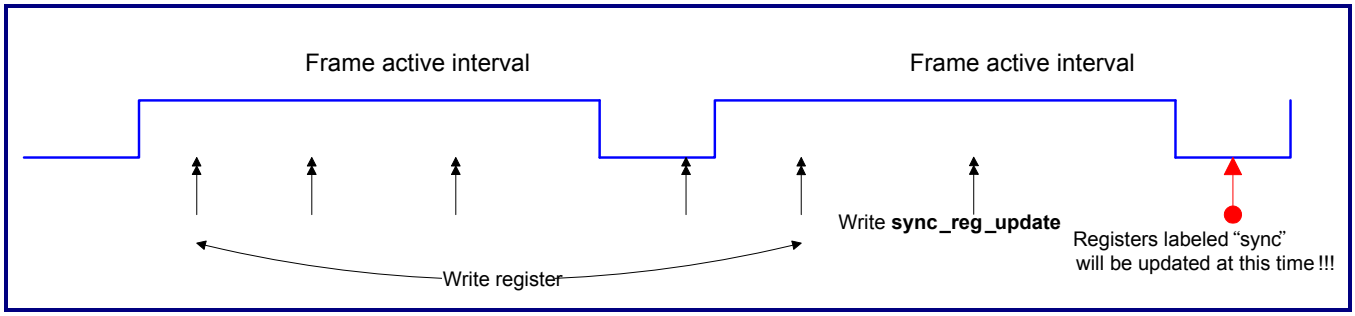



**Figure 19-2. Timing of sync_reg_update**

### 19.4.2.3 CAPCLK (Offset: 0x0008)

**Table 19-4. CAPCLK Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| 28 | SW_RST | R/W | 0 | Software reset (Auto clear)<br>1: Reset |
| [27:17] | - | - | - | Reserved |
| 16 | INT_SYNC_CLR | R/W | 0 | Automatically clear the interrupt source at every vertical start<br>0: Manually clear<br>1: Automatically clear |
| [15:13] | - | - | - | Reserved |
| 12 | ASST_CLK_SEL | R/W | 0 | Select external assistant clock<br>0: Select PCLK as the main clock<br>1: Select the external assistant clock, vcap_iclk, as the main clock |
| [11:5] | - | - | - | - |
| 4 | VP_CLK_INV | R/W | 0 | Invert the input video pixel clock<br>0: Disable<br>1: Enable |
| [3:0] | - | - | - | Reserved |

### 19.4.2.4 P0SIZE0 (Offset: 0x0010)

**Table 19-5. PVSIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P0_TARGET_HEIGHT | R/W | 0 | Height of an output target image for Path 0 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P0_TARGET_WIDTH | R/W | 0 | Width of an output target image in multiples of 4 for Path 0 (Sync. register) |

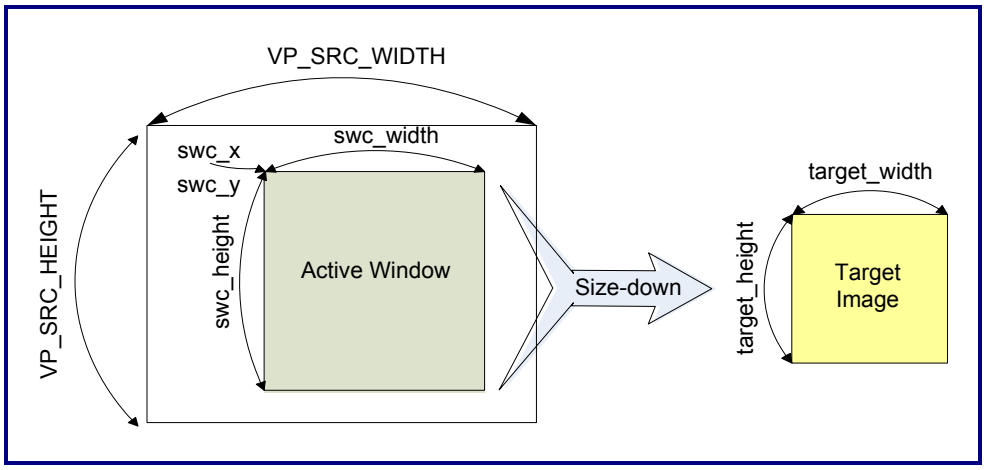



**Figure 19-3. Relationship between Source Image and Target Image**

### 19.4.2.5 P0SIZE1 (Offset: 0x0014)

**Table 19-6. PVSIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P0_CROP_Y | R/W | 00 | Y-coordinate at the top-left corner for the output crop window of Path 0 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P0_CROP_X | R/W | - | X-coordinate at the top-left corner for the output crop window of Path 0 (Sync. register) |

### 19.4.2.6 P0SIZE2 (Offset: 0x0018)

**Table 19-7. PVSIZE2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P0_CROP_HEIGHT | R/W | 0 | Height of an output crop image in multiples of 2 for Path 0 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P0_CROP_WIDTH | R/W | 0 | Width of an output crop image in multiples of 4 for Path 0 (Sync. register) |

### 19.4.3 P1SHARPEN (Offset 0x00D8)

**Table 19-8. P1SHARPEN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | P1_H_BLANK_PIX | R/W | 8 | Width of horizontal blank for the extended size-up line on Path 1 |
| [23:10] | - | - | - | Reserved |
| [9:8] | P1_ASPECT_MAJOR | R/W | 0 | Aspect ratio control for image resize on Path 1<br>00: Resize the output image in the X-, Y-direction independent mode<br>01: Resize the output image to keep the aspect ratio by X-direction major<br>10: Resize the output image to keep the aspect ratio by Y-direction major<br>11: Reserved |
| [7:0] | - | - | - | Reserved |

### 19.4.4 P1SIZE0 (Offset 0x0020)

**Table 19-9. P1SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P1_TARGET_HEIGHT | R/W | 0 | Height of an output target image for Path 1 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P1_TARGET_WIDTH | R/W | 0 | Width of an output target image in multiples of 4 for Path 1 (Sync. register) |

### 19.4.5 P1SIZE1 (Offset 0x0024)

**Table 19-10. P1SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P1_CROP_Y | R/W | 0 | Y-coordinate at the top-left corner for the output crop window of Path 1 |
| [15:12] | - | - | - | Reserved |
| [11:0] | P1_CROP_X | R/W | 0 | X-coordinate at the top-left corner for the output crop window of Path 1 |

## 19.4.6 P1SIZE2 (Offset 0x0028)

**Table 19-11. P1SIZE2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P1_CROP_HEIGHT | R/W | 0 | Height of an output crop image in multiples of 2 for path 1 |
| [15:12] | - | - | - | Reserved |
| [11:0] | P1_CROP_WIDTH | R/W | 0 | Width of an output crop image in multiples of 4 for path 1 |

## 19.4.7 DFORMAT Register (Offset 0x0038)

**Table 19-12. DFORMAT Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:14] | - | - | - | Reserved |
| 13 | P1_DEST_YC_SWAP | R/W | 0 | Destination CbCr/Y sequence swap for path 1.<br>This bit is effective if DMA write out YCbCr data into memory by packed format<br>0: Sequence of CbCr is ahead of Y (Cb Y Cr or Cr Y Cb)<br>1: Sequence of Y is ahead of CbCr (Y Cb Cr or Y Cr Cb) |
| 12 | P1_DEST_CBCR_SWAP | R/W | 0 | Destination Cr/Cb sequences swap for Path 1<br>This bit will be effective if DMA writes out the YCbCr data into the memory by packed forma.<br>0: Sequence of Cr is ahead of Cb → Cr Cb.<br>1: Sequence of Cb is ahead of Cr → Cb Cr. |
| [11:8] | P1_DEST_FORMAT | R/W | 0 | Data output format for Path 1<br>x000 ~ x100: Reserved<br>x101: YCbCr 4:2:2<br>0xxx: YCbCr 256 level<br>1xxx: YCbCr 240 level |
| [7:6] | - | - | - | Reserved |
| 5 | P0_DEST_YC_SWAP | R/W | 0 | Destination CbCr/Y sequence swap for path 0<br>This bit will be effective if DMA writes out the YCbCr data into the memory by packed forma.<br>0: Sequence of CbCr is ahead of Y (Cb Y Cr or Cr Y Cb).<br>1: Sequence of Y is ahead of CbCr (Y Cb Cr or Y Cr Cb). |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 4 | P0_DEST_CBCR_SWAP | R/W | 0 | Destination Cr/Cb sequences swap for Path 0<br>This bit will be effective if DMA writes out the YCbCr data into the memory by packed forma.<br>0: Sequence of Cr is ahead of Cb → CrCb.<br>1: Sequence of Cb is ahead of Cr → CbCr. |
| [3:0] | P0_DEST_FORMAT | R/W | 0 | Data output format for Path 0<br>x000 ~ x100: Reserved<br>x101: YCbCr 4:2:2<br>0xxx: YCbCr 256 level<br>1xxx: YCbCr 240 level |

### 19.4.8 GBATRB (Offset 0x003C)

**Table 19-13. GBATRB Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:20] | - | - | - | Reserved |
| 19 | FIELD_ATRB_EN | R/W | 0 | VCAP to recognize field attribute or not |

| field_atrb_en | field_pair_en | field_seq | action |
|---|---|---|---|
| 0 | 0 | x | Any field/frame |
| 1 | 0 | 0 | Odd field only |
| 1 | 0 | 1 | Even field only |
| 0 | 1 | x | Any two fields |
| 1 | 1 | 0 | Odd then even |
| 1 | 1 | 1 | Even then odd |

| | | | | |
|---|---|---|---|---|
| 18 | - | - | - | Reserved |
| 17 | FIELD_PAIR_EN | R/W | 0 | Get one field/frame or field-pair per fire of one stepping capture<br>0: One field/frame capture<br>1: Get one field pair per fire of one stepping-capture |
| 16 | FIELD_SEQ | R/W | 0 | Field capture sequence by odd or even first<br>0: Odd first<br>1: Even first |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [15:14] | P1_SWC_SRC_SEL | R/W | 0 | SWC source selection for path 1<br>(Sync register)<br>0: Select the video data after DI<br>1: Select the video data before DI<br>2, 3: Select the video data from another video group by local video bus |
| [13:12 | - | - | - | Reserved |
| 11 | P1_OSD_FM_FORCE | R/W | 1 | Force path 1 OSD frame type<br>0: Depend on front-end frame/field type<br>1: Force the OSD frame type to the progressive mode regardless of front-end frame/field type |
| 10 | P0_OSD_FM_FORCE | R/W | 0 | Force path 0 OSD frame type<br>0: Depend on front-end frame/field type<br>1: Force the OSD frame type to the progressive mode regardless of front-end frame/field type |
| 9 | SWC_CENTER_XY | R/W | 1 | Force source active window to locate at center of image<br>Do not care the start point or SWC_X and SWC_Y<br>(Sync register)<br>0: Disable<br>1:Enable |
| 8 | VP_RANGE | R/W | 0 | Data range of input video<br>0: 16 ← Y ← 235<br>16 ← Cb/Cr ← 240<br>1: 1 ← Y/Cb/Cr ← 254 |
| [7:6] | P0_SWC_SRC_SEL | R/W | 0 | SWC source selection for path 0<br>(Sync register)<br>0: Select the video data after DI<br>1: Select the video data before DI<br>2, 3: Select the video data from another video group by local video bus |
| [5:4] | - | - | - | Reserved |
| 3 | IMG_SRC_SEL | R/W | 0 | Global source image selection<br>0: Source image is from the external image/video.<br>1: Source image is from the memory. |
| 2 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [1:0] | VP_SRC_SEL | R/W | 0 | Video port source selection<br>00: Select the video port from data input 0<br>01: Select the video port from data input 1<br>10: Select the video port from data input 2<br>11: Reserved |

### 19.4.9 FMRATE (Offset 0x0040)

**Table 19-14. FMRATE Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | P1_CAP_FIRE | R/W | 0 | Fire to capture frame on path zero by the method specified in bits[11:8](p0_field_xxx)<br>This bit will be set after triggered by users and will be kept until the end of specified frame capture (Auto clear).<br>0: Fire one frame capture action on path 1 |
| [30:28] | - | - | - | Reserved |
| 27 | P1_FED_ATRB_EN | R/W | 0 | Path 1 VCAP to recognize field attribute or not<br>0: Ignore the field attribute<br>1: Enable the capture to recognize the field attribute |

| field_atrb_en | field_pair_en | field_seq | action |
|---|---|---|---|
| 0 | 0 | x | Any one field/frame |
| 1 | 0 | 0 | Odd field only |
| 1 | 0 | 1 | Even field only |
| 0 | 1 | x | Any two field |
| 1 | 1 | 0 | Odd then even |
| 1 | 1 | 1 | Even then odd |

| | | | | |
|---|---|---|---|---|
| 26 | - | - | - | Reserved |
| 25 | P1_FIELD_PAIR_EN | R/W | 0 | Get one field/frame or field-pair per fire of one stepping capture<br>0: One field/frame capture<br>1: Get one field pair per fire of one stepping-capture |
| 24 | P1_FIELD_SEQ | R/W | 0 | Path 1 VCAP sequence by odd or even first<br>0: Odd first<br>1: Even first |
| [23:21] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [20:16] | P1_FM_RATE | R/W | 0 | The frame interval for path 1 to pass one frame<br>(P1 sync reg)<br>0: Do not skip any frame<br>1 ~ 30: Number of skipping-frame interval<br>31: Block all frame |
| 15 | P0_CAP_FIRE | R/W | 0 | Fire to capture frame on path zero by the method specified in bits[11:8] (p0_field_xxx)<br>This bit will be set after triggered by users and will be kept until the end of specified frame capture (Auto clear).<br>0: Fire one frame capture action on path 0 |
| [14:12] | - | - | - | Reserved |
| 11 | P0_FED_ATRB_EN | R/W | 0 | Path 0 VCAP to recognize field attribute or not<br>0: Ignore the field attribute<br>1: Enable the capture to recognize the field attribute |
| 10 | - | - | - | Reserved |
| 9 | P0_FIELD_PAIR_EN | R/W | 0 | Get one field/frame or field-pair per fire of one stepping capture<br>0: One field/frame capture<br>1: Get one field pair per fire of one stepping-capture |
| 8 | P0_FIELD_SEQ | R/W | 0 | Path 0 VCAP sequence by odd or even first<br>0: Odd first<br>1: Even first |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_FM_RATE | R/W | 0 | The frame interval for path 0 to pass one frame<br>(P0 sync reg)<br>0: Do not skip any frame<br>1 ~ 30: Number of skipping-frame interval<br>31: Block all frame |

| field_atrb_en | field_pair_en | field_seq | action |
|---|---|---|---|
| 0 | 0 | x | Any one field/frame |
| 1 | 0 | 0 | Odd field only |
| 1 | 0 | 1 | Even field only |
| 0 | 1 | x | Any two field |
| 1 | 1 | 0 | Odd then even |
| 1 | 1 | 1 | Even then odd |

### 19.4.10 SRCIF (Offset 0x0044)

**Table 19-15. SRCIF Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| 28 | YUV16_CBCR_SWAP | R/W | 0 | Swap Cb, Cr input sequence |
| [27:21] | - | - | - | Reserved |
| 20 | OSD_FIELD_CTRL | | | Both path0 and path 1 OSD field sequence control.<br>0: Field type inverse<br>1: Field type non-inverse |
| 19 | VP_PROG | R/W | 0 | Define Video input as progressive or interlace mode<br>0: Interlace mode<br>1: Progressive mode |
| [18:16] | - | - | - | Reserved |
| 15 | VP_H_REF_POL | R/W | 1 | Active polarity of video-port horizontal sync./valid control<br>0: Video-port horizontal control timing is active low.<br>1: Video-port horizontal control timing is active high. |
| 14 | VP_V_REF_POL | R/W | 1 | Active polarity of video port vertical sync./valid control<br>0: Video-port vertical control timing is active low.<br>1: Video-port vertical control timing is active high. |
| 13 | VP_H_REF_DEF | R/W | 0 | The definition of video port horizontal sync./valid control<br>0: Video-port horizontal control timing is treated as data valid signal.<br>1: Video-port horizontal control timing is treated as sync. signal. |
| 12 | VP_V_REF_DEF | R/W | 0 | The definition of video port vertical sync./valid control<br>0: Video-port vertical control timing is treated as data valid signal.<br>1: Video port Vertical control timing is treated as synchronization signal. |
| 11 | VP_SYNC_EMB | R/W | 1 | H/V sync. embedded in or separated from video stream<br>0: Sync. signal are separated from video data.<br>1: Sync. signal are embedded in video data. |
| 10 | - | - | - | Reserved |
| [9:8] | VP_IF_FORMAT | R/W | 0 | Video port interface format<br>00: CCIR-656 format<br>01: Reserved<br>10: SONY 16-bit YUV format<br>11: General 16-bit YCbCr format |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [7:0] | - | - | - | Reserved |

## 19.4.11 SRCSIZE0 (Offset 0x0050)

**Table 19-16. SRCSIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | VP_SRC_HEIGHT | R/W | 0 | Video port image source height (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | VP_SRC_WIDTH | R/W | 0 | Video port image source width (Sync. register) |

## 19.4.12 P0SIZE3 (Offset 0x0054)

**Table 19-17. P0SIZE3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P0_SWC_Y | R/W | 0 | Y-coordinate of top-left corner for source active window on path 0<br>In multiples of 2 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P0_SWC_X | R/W | 0 | X-coordinate of top-left corner for source active window on path 0<br>In multiples of 2 (Sync. register) |

## 19.4.13 P0SIZE4 (Offset 0x0058)

**Table 19-18. P0SIZE4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P0_SWC_HEIGHT | R/W | 0 | Image source active window height on path 0<br>In multiples of 2 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P0_SWC_WIDTH | R/W | 0 | Image source active window width on path 0<br>In multiples of 4 (Sync. register) |

## 19.4.14 P0SIZE5 (Offset 0x005C)

**Table 19-19. P0SIZE5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P0_SD_HEIGHT | R | 0 | Image width after sizing down on path 0 |
| [15:12] | - | - | - | Reserved |
| [11:0] | P0_SD_WIDTH | R | 0 | Image height after sizing down on path 0. |

## 19.4.15 SRCSIZE1 (Offset 0x0060)

**Table 19-20. SRCSIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | VP_SRC_Y | R/W | 0x05 | Vertical pixel capture start point of source image |
| [15:12] | - | - | - | Reserved |
| [11:0] | VP_SRC_X | R/W | 0x38 | Horizontal pixel capture start point of source image |

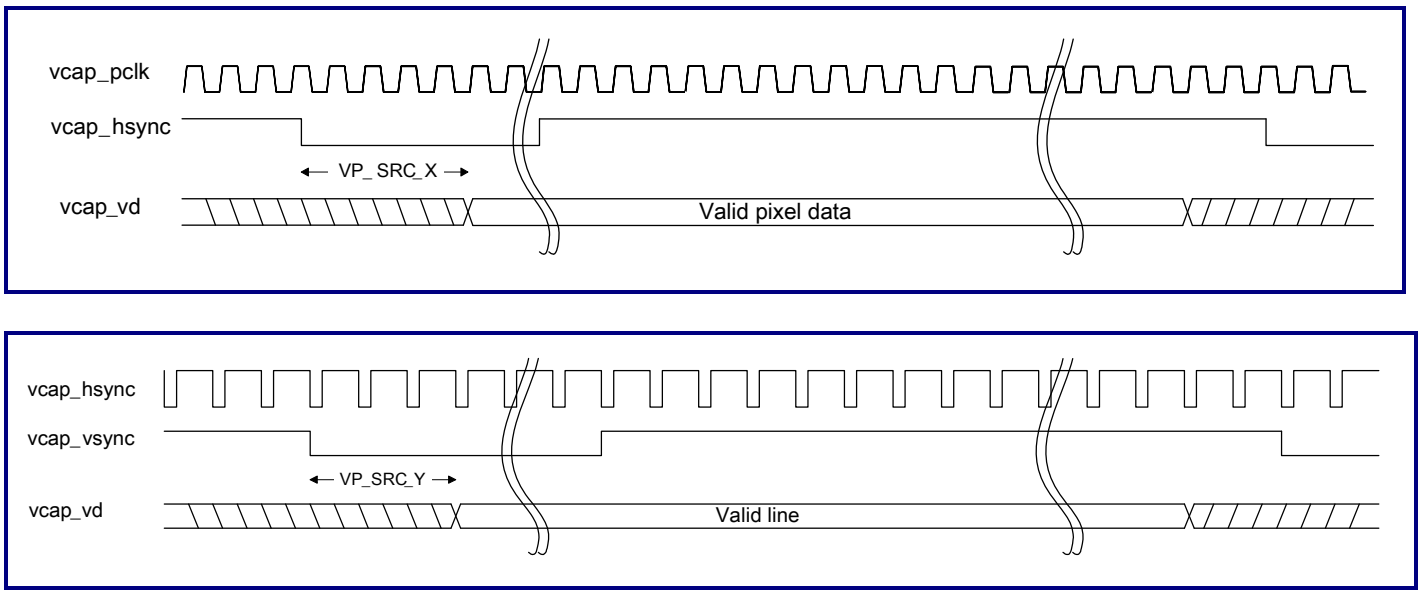



**Figure 19-4. VPSRC_X and VP_SRC_Y**

## 19.4.16 P1SIZE3 (Offset 0x0064)

**Table 19-21. P1SIZE3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P1_SWC_Y | R/W | 00 | Y-coordinate of top-left corner for source active window on path 1<br>In multiples of 2 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P1_SWC_X | R/W | - | X-coordinate of top-left corner for source active window on path 1<br>In multiples of 2 (Sync. register) |

## 19.4.17 P1SIZE4 (Offset 0x0068)

**Table 19-22. P1SIZE4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P1_SWC_HEIGHT | R/W | 0 | Image source active window height on path 1.<br>In multiples of 2 (Sync. register) |
| [15:12] | - | - | - | Reserved |
| [11:0] | P1_SWC_WIDTH | R/W | 0 | Image source active window width on path 1.<br>In multiples of 4 (Sync. register) |

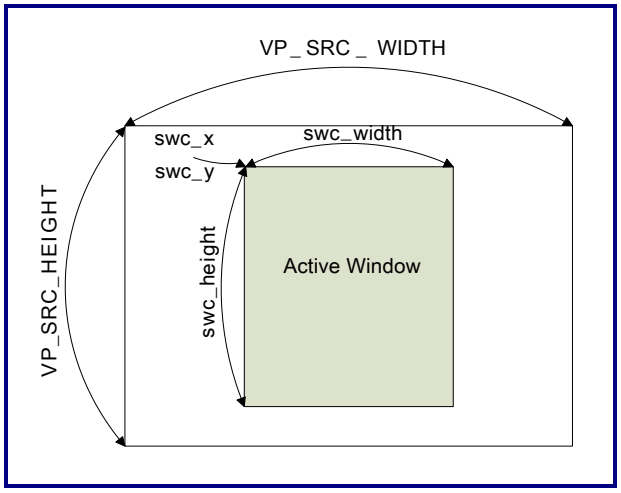



**Figure 19-5. Source Active Window**

## 19.4.18 P1SIZE5 (Offset 0x006C)

**Table 19-23. P1SIZE5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | P1_SD_HEIGHT | R | 0 | Image width after sizing down on path 1 |
| [15:12] | - | - | - | Reserved |
| [11:0] | P1_SD_WIDTH | R | 0 | Image height after sizing down on path 1. |

## 19.4.19 P1SHARPEN (Offset 0x00D8)

**Table 19-24. P1SHARPEN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | P1_SP_K2 | | | Sharpness gain factor 2 on path 1.<br>Format: 2.2 |
| [23:20] | - | - | - | Reserved |
| [19:16] | P1_SP_K1 | | | Sharpness gain factor 1 on path 1<br>Format: 2.2 |
| 15 | - | - | - | Reserved |
| [14:8] | P1_SP_TH2 | R/W | 30 | Sharpness threshold 2 on path 1<br>Format 7 |
| 7 | - | - | - | Reserved |
| [6:0] | P1_SP_TH1 | R/W | 30 | Sharpness threshold 1 on path 1<br>Format 7 |

## 19.4.20 P0MSK0COR (Offset 0x0100)

**Table 19-25. P0MSK0COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [6:4] | P0_MSK_WIN0_COR | R/W | 0x00 | Mask window 0 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P0_MSK_WIN0_TRAN | R/W | 0x00 | Mask window 0 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.21 P0MSK0SIZE0 (Offset 0x0104)

**Table 19-26. P0MSK0SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN0_Y_START | - | - | Start line of mask window 0 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN0_X_START | - | - | Start pixel of mask window 0 |

## 19.4.22 P0MSK0SIZE1 (Offset 0x0108)

**Table 19-27. P0MSK0SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN0_Y_END | - | - | End line of mask window 0 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN0_X_END | - | - | End pixel of mask window 0 |

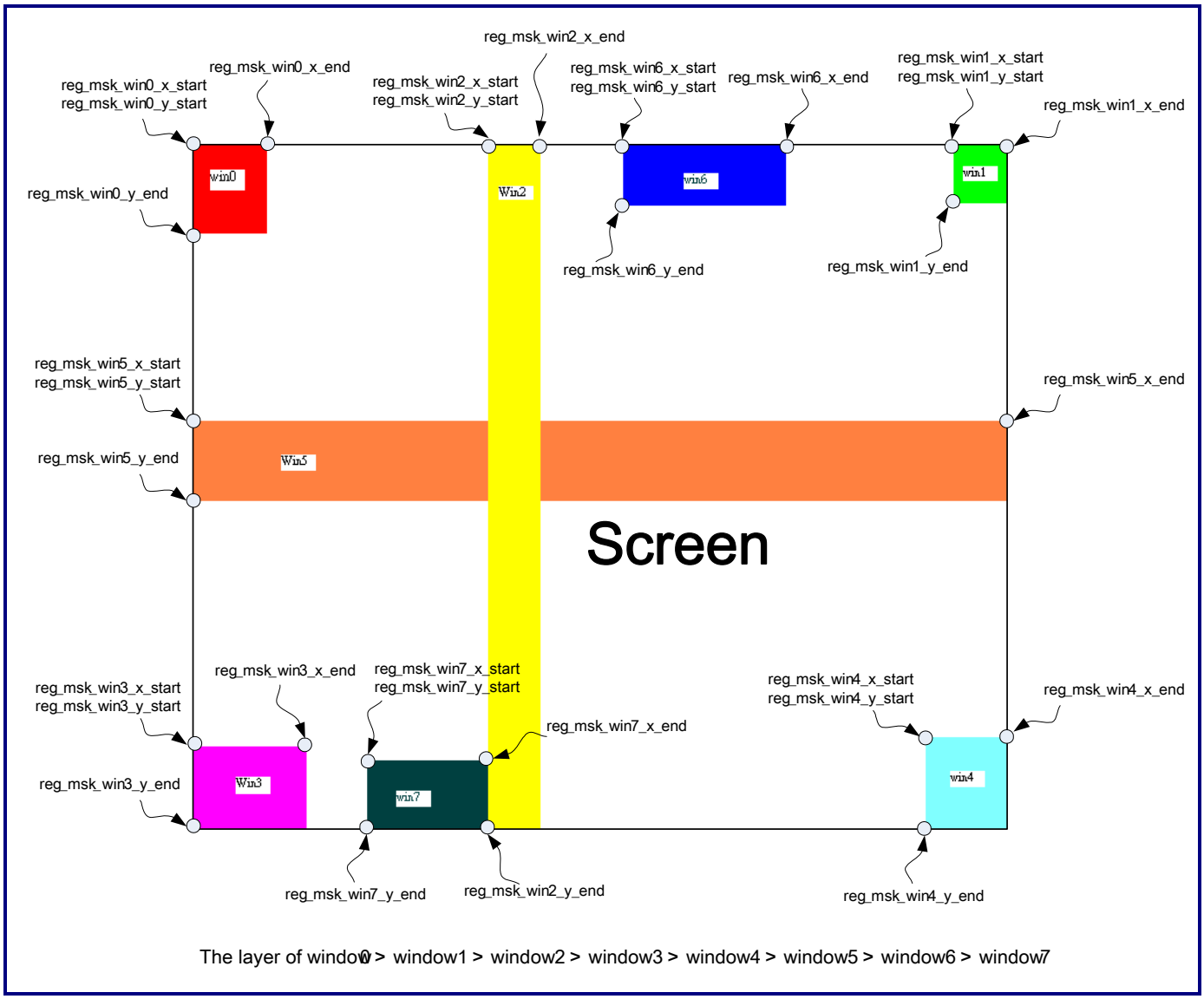



**Figure 19-6. Window Mask**

## 19.4.23 P0MSK1COR (Offset 0x010C)

**Table 19-28. P0MSK1COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P0_MSK_WIN1_COR | R/W | 0x00 | Mask window 1 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P0_MSK_WIN1_TRAN | R/W | 0x00 | Mask window 1 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.24 P0MSK1SIZE0 (Offset 0x0110)

**Table 19-29. P0MSK1SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN1_Y_START | - | - | Start line of mask window 1 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN1_X_START | - | - | Start pixel of mask window 1 |

## 19.4.25 P0MSK1SIZE1 (Offset 0x0114)

**Table 19-30. P0MSK1SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN1_Y_END | | | End line of mask window 1 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN1_X_END | | | End pixel of mask window 1 |

## 19.4.26 P0MSK2COR (Offset 0x0118)

**Table 19-31. P0MSK2COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P0_MSK_WIN2_COR | R/W | 0x00 | Mask window 2 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P0_MSK_WIN2_TRAN | R/W | 0x00 | Mask window 2 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.27 P0MSK2SIZE0 (Offset 0x011C)

**Table 19-32. P0MSK2SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN2_Y_START | | | Start line of mask window 2 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN2_X_START | | | Start pixel of mask window 2 |

## 19.4.28 P0MSK2SIZE1 (Offset 0x0120)

**Table 19-33. P0MSK2SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN2_Y_END | | | End line of mask window 2 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN2_X_END | | | End pixel of mask window 2 |

## 19.4.29 P0MSK1COR (Offset 0x0124)

**Table 19-34. P0MSK3COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P0_MSK_WIN3_COR | R/W | 0x00 | Mask window 3 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [1:0] | P0_MSK_WIN3_TRAN | R/W | 0x00 | Mask window 3 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.30 P0MSK3SIZE0 (Offset 0x0128)

**Table 19-35. P0MSK3SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN3_Y_START | | | Start line of mask window 3 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN3_X_START | | | Start pixel of mask window 3 |

## 19.4.31 P0MSK3SIZE1 (Offset 0x012C)

**Table 19-36. P0MSK3SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN3_Y_END | | | End line of mask window 3 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN3_X_END | | | End pixel of mask window 3 |

## 19.4.32 P0MSK4COR (Offset 0x0130)

**Table 19-37. P0MSK4COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P0_MSK_WIN4_COR | R/W | 0x00 | Mask window 4 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P0_MSK_WIN4_TRAN | R/W | 0x00 | Mask window 4 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.33 P0MSK4SIZE0 (Offset 0x0134)

**Table 19-38. P0MSK4SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN4_Y_START | | | Start line of mask window 4 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN4_X_START | | | Start pixel of mask window 4 |

## 19.4.34 P0MSK4SIZE1 (Offset 0x0138)

**Table 19-39. P0MSK4SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN4_Y_END | | | End line of mask window 4 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN4_X_END | | | End pixel of mask window 4 |

## 19.4.35 P0MSK5COR (Offset 0x013C)

**Table 19-40. P0MSK5COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P0_MSK_WIN5_COR | R/W | 0x00 | Mask window 5 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P0_MSK_WIN5_TRAN | R/W | 0x00 | Mask window 5 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.36 P0MSK5SIZE0 (Offset 0x0140)

**Table 19-41. P0MSK0SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN5_Y_START | | | Start line of mask window 5 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN5_X_START | | | Start pixel of mask window 5 |

## 19.4.37 P0MSK5SIZE1 (Offset 0x0144)

**Table 19-42. P0MSK5SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN5_Y_END | | | End line of mask window 5 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN5_X_END | | | End pixel of mask window 5 |

## 19.4.38 P0MSK6COR (Offset 0x0148)

**Table 19-43. P0MSK6COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P0_MSK_WIN6_COR | R/W | 0x00 | Mask window 6 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [1:0] | P0_MSK_WIN6_TRAN | R/W | 0x00 | Mask window 6 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.39 P0MSK6SIZE0 (Offset 0x014C)

**Table 19-44. P0MSK06SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN6_Y_START | - | - | Start line of mask window 6 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN6_X_START | - | - | Start pixel of mask window 6 |

## 19.4.40 P0MSK6SIZE1 (Offset 0x0150)

**Table 19-45. P0MSK6SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN6_Y_END | - | - | Mask the end line of window 6 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN6_X_END | - | - | Mask the end pixel of window 6 |

## 19.4.41 P0MSK6COR (Offset 0x0154)

**Table 19-46. P0MSK6COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P0_MSK_WIN7_COR | R/W | 0x00 | Mask the color index of window 7<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P0_MSK_WIN7_TRAN | R/W | 0x00 | Mask the color transparency level of window 7<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.42 P0MSK7SIZE0 (Offset 0x0158)

**Table 19-47. P0MSK7SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN7_Y_START | - | - | Mask the start line of window 7 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN7_X_START | - | - | Mask the start pixel of window 7 |

## 19.4.43 P0MSK7SIZE1 (Offset 0x015C)

**Table 19-48. P0MSK7SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P0_MSK_WIN7_Y_END | - | - | Mask the end line of window 7 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_MSK_WIN7_X_END | - | - | Mask the end pixel of window 7 |

## 19.4.44 P1MSK0COR (Offset 0x0160)

**Table 19-49. P1MSK0COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P1_MSK_WIN0_COR | R/W | 0x00 | Mask the color index of window 0<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P1_MSK_WIN0_TRAN | R/W | 0x00 | Mask the color transparency level of window 0<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.45 P1MSK0SIZE0 (Offset 0x0164)

**Table 19-50. P1MSK0SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN0_Y_START | - | - | Mask the start line of window 0 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN0_X_START | - | - | Mask the start pixel of window 0 |

## 19.4.46 P1MSK0SIZE1 (Offset 0x0168)

**Table 19-51. P1MSK0SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN0_Y_END | - | - | Mask the end line of window 0 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN0_X_END | - | - | Mask the end pixel of window 0 |

## 19.4.47 P1MSK1COR (Offset 0x016C)

**Table 19-52. P1MSK1COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P1_MSK_WIN1_COR | R/W | 0x00 | Mask the color index of window 1<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [1:0] | P1_MSK_WIN1_TRAN | R/W | 0x00 | Mask the color transparency level of window 1<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.48 P1MSK1SIZE0 (Offset 0x0170)

**Table 19-53. P1MSK1SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN1_Y_START | - | - | Mask the start line of window 1 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN1_X_START | - | - | Mask the start pixel of window 1 |

## 19.4.49 P1MSK1SIZE1 (Offset 0x0174)

**Table 19-54. P1MSK1SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN1_Y_END | - | - | Mask the end line of window 1 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN1_X_END | - | - | Mask the end pixel of window 1 |

## 19.4.50 P1MSK2COR (Offset 0x0178)

**Table 19-55. P1MSK2COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P1_MSK_WIN2_COR | R/W | 0x00 | Mask the color index of window 2<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P1_MSK_WIN2_TRAN | R/W | 0x00 | Mask the color transparency level of window 2<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.51 P1MSK2SIZE0 (Offset 0x017C)

**Table 19-56. P1MSK2SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN2_Y_START | - | - | Mask the start line of window 2 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN2_X_START | - | - | Mask the start pixel of window 2 |

## 19.4.52 P1MSK2SIZE1 (Offset 0x0180)

**Table 19-57. P1MSK2SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN2_Y_END | - | - | Mask the end line of window 2 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN2_X_END | - | - | Mask the end pixel of window 2 |

## 19.4.53 P1MSK1COR (Offset 0x0184)

**Table 19-58. P1MSK3COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31-7 | - | - | - | Reserved |
| 6-4 | P1_MSK_WIN3_COR | R/W | 0x00 | Mask the color index of window 3<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| 3-2 | - | - | - | Reserved |
| 1-0 | P1_MSK_WIN3_TRAN | R/W | 0x00 | Mask the color transparency level of window 3<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.54 P1MSK3SIZE0 (Offset 0x0188)

**Table 19-59. P1MSK3SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN3_Y_START | - | - | Mask the start line of window 3 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN3_X_START | - | - | Make the start pixel of window 3 |

## 19.4.55 P1MSK3SIZE1 (Offset 0x018C)

**Table 19-60. P1MSK3SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN3_Y_END | - | - | Mask the end line of window 3 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN3_X_END | - | - | Mask the end pixel of window 3 |

## 19.4.56 P1MSK4COR (Offset 0x0190)

**Table 19-61. P1MSK4COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31-7 | - | - | - | Reserved |
| 6-4 | P1_MSK_WIN4_COR | R/W | 0x00 | Mask the color index of window 4<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| 3-2 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 1-0 | P1_MSK_WIN4_TRAN | R/W | 0x00 | Mask the color transparency level of window 4<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.57 P1MSK4SIZE0 (Offset 0x0194)

**Table 19-62. P1MSK4SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN4_Y_START | - | - | Mask the start line of window 4 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN4_X_START | - | - | Mask the start pixel of window 4 |

## 19.4.58 P1MSK4SIZE1 (Offset 0x0198)

**Table 19-63. P1MSK4SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN4_Y_END | - | - | Mask the end line of window 4 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN4_X_END | - | - | Mask the end pixel of window 4 |

## 19.4.59 P1MSK5COR (Offset 0x019C)

**Table 19-64. P1MSK5COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P1_MSK_WIN5_COR | R/W | 0x00 | Mask the color index of window 5<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P1_MSK_WIN5_TRAN | R/W | 0x00 | Mask the color transparency level of window 5<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.60 P1MSK5SIZE0 (Offset 0x01A0)

**Table 19-65. P1MSK0SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN5_Y_START | - | - | Mask the start line of window 5 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN5_X_START | - | - | Mask the start pixel of window 5 |

## 19.4.61 P1MSK5SIZE1 (Offset 0x01A4)

**Table 19-66. P1MSK5SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN5_Y_END | - | - | Mask the end line of window 5 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN5_X_END | - | - | Mask the end pixel of window 5 |

## 19.4.62 P1MSK6COR (Offset 0x01A8)

**Table 19-67. P1MSK6COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P1_MSK_WIN6_COR | R/W | 0x00 | Mask the color index of window 6<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |
| [1:0] | P1_MSK_WIN6_TRAN | R/W | 0x00 | Mask the color transparency level of window 6<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.63 P1MSK6SIZE0 (Offset 0x01AC)

**Table 19-68. P1MSK06SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN6_Y_START | - | - | Start line of mask window 6 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN6_X_START | - | - | Start pixel of mask window 6 |

## 19.4.64 P1MSK6SIZE1 (Offset 0x01B0)

**Table 19-69. P1MSK6SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN6_Y_END | - | - | End line of mask window 6 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN6_X_END | - | - | End pixel of mask window 6 |

## 19.4.65 P1MSK6COR (Offset 0x01B4)

**Table 19-70. P1MSK6COR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:7] | - | - | - | Reserved |
| [6:4] | P1_MSK_WIN7_COR | R/W | 0x00 | Mask window 7 color index<br>0: Palette color No. 0<br>1: Palette color No. 1<br>2: Palette color No. 2<br>3: Palette color No. 3<br>4: Palette color No. 4<br>5: Palette color No. 5<br>6: Palette color No. 6<br>7: Inverse background color |
| [3:2] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [1:0] | P1_MSK_WIN7_TRAN | R/W | 0x00 | Mask window 7 color transparency level<br>0: 0% (Normal)<br>1: 50%<br>2: 75%<br>3: 100% |

## 19.4.66 P1MSK7SIZE0 (Offset 0x01B8)

**Table 19-71. P1MSK7SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN7_Y_START | - | - | Start line of mask window 7 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN7_X_START | - | - | Start pixel of mask window 7 |

## 19.4.67 P1MSK7SIZE1 (Offset 0x01BC)

**Table 19-72. P1MSK7SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | P1_MSK_WIN7_Y_END | - | - | End line of mask window 7 |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_MSK_WIN7_X_END | - | - | End pixel of mask window 7 |

## 19.4.68 DICTRL0 (Offset 0x01E0)

**Table 19-73. DICTRL0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | DI_ELA_H_TH | R/W | 0x3C | High threshold for ELA value calculation |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [23:16] | DI_ELA_L_TH | R/W | 0x14 | Low threshold for ELA value calculation<br>High/Low threshold are used to correct the result of the interpolation<br>For new ELA method, DI_ELA_H_TH should be greater than or equal to DI_ELA_L_TH . (DI_ELA_H_TH >= DI_ELA_L_TH)<br>When DI_ELA_H_TH < DI_ELA_L_TH, ELA will become old version.<br>New ELA approach is recommended. |
| [15:3] | - | - | - | Reserved |
| [2:0] | DI_WEIGHT_TH | R/W | 0x05 | Weighted judgment threshold<br>Used to determine the pixel interpolation direction<br>Only when DI_ELA_H_TH < DI_ELA_L_TH, DI_WEIGHT_TH will effect ELA result.<br>Otherwise, this threshold is calculated inherently. |

## 19.4.69 DICTRL1 (Offset 0x001E4)

**Table 19-74. DICTRL1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:8] | - | - | - | Reserved |
| [7:0] | DI_H_BLANK_PIX | R/W | 0x0F | Horizontal blank interval between intra and odd/even line |

## 19.4.70 MISC0 (Offset 0x001F0)

**Table 19-75. MISC0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:18] | - | - | - | Reserved |
| 17 | P1_SD_CALC_ACU | R/W | 0x0 | Path-1 intermediate value of size down calculation accuracy control. When size down ratio larger than 4, size down internal operation will use 16-bit precision for one pixel.<br>0: Disable<br>1: Enable |
| 16 | P0_SD_CALC_ACU | R/W | 0x0 | Path-0 intermediate value of size down calculation accuracy control. When size down ratio larger than 4, size down internal operation will use 16-bit precision for one pixel.<br>0: Disable<br>1: Enable |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [15:0] | - | - | - | Reserved |

## 19.4.71 P0DMA0 (Offset 0x0200)

**Table 19-76. PDMA0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | - | - | - | Reserved |
| [30:28] | P0DMA_WM | R/W | 4 | Low-level watermark of available DMA FIFO space on path 0 |
| 27 | | - | - | Reserved |
| [26:24] | P1_HRESP_DLY | R/W | 0 | AHB retry/split delay clock = 8+ pv_hresp_dly<br>For path 0 |
| [23:17] | - | - | - | Reserved |
| 16 | P0DMA_MST_EN | R/W | 0 | Multi-start address enable for Path-0 DMA write<br>0: Enable the frame numbers 0 start address<br>1: Enable the frame numbers 0, 1 start address |
| 15 | P0DMA_MJMP | - | - | Perform the path 0 DMA ordering by jumping to the next location in distance n per n pixel for n x n macro block order. N = 8, 16<br>0: Disable<br>1: Enable |
| 14 | - | - | - | Reserved |
| [13:12] | P0DMA_ORDER | R/W | 0 | YCbCr 4:2:2 data organization in three memory buffers for path 0<br>00: Sequential order (Supported when DMA_SPLIT = 0)<br>Others: Reserved |
| 11 | P0_DMA_V_FLIP | R/W | 0 | Up-stream DMA vertical flipping mode switch on path 0.<br>Different destination base addresses need to be programmed for each flipping mode. 0: No flipping on the DMA vertical direction<br>1: Flipping on the DMA vertical direction |
| 10 | P0_DMA_H_FLIP | R/W | 0 | Up-stream DMA horizontal flipping mode switch on path 0.<br>Different destination base addresses need to be programmed for each flipping mode.<br>0: No flipping on the DMA horizontal direction<br>1: Flipping on the DMA horizontal direction |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [9:8] | P0DMA_SPLIT | R/W | 0 | YCbCr data arrangement in memory<br>00: Write three color components into the single memory block (Packed)<br>Others: Reserved |
| [7:6] | P0DMA_C_WLEN | R/W | 0 | Cb/Cr burst write length<br>00: 1 data package for one burst<br>01: 2 data package for one burst<br>10: 4 data package for one burst<br>11: 8 data package for one burst |
| [5:4] | P0DMA_Y_WLEN | R/W | 0 | Y burst write length<br>00: 4 data package for one burst<br>01: 8 data package for one burst<br>10: 12 data package for one burst<br>11: 16 data package for one burst |
| [3:2] | P0DMA_RLEN | R/W | 0 | DMA read burst length on path 0<br>00: 8 data package for one burst<br>01: 16 data package for one burst<br>10 ~ 11: Reserved |
| [1:0] | P0DMA_WLEN | R/W | 0 | DMA write burst length on path 0<br>00: 8 data package for one burst<br>01: 16 data package for one burst<br>10 ~ 11: Reserved |

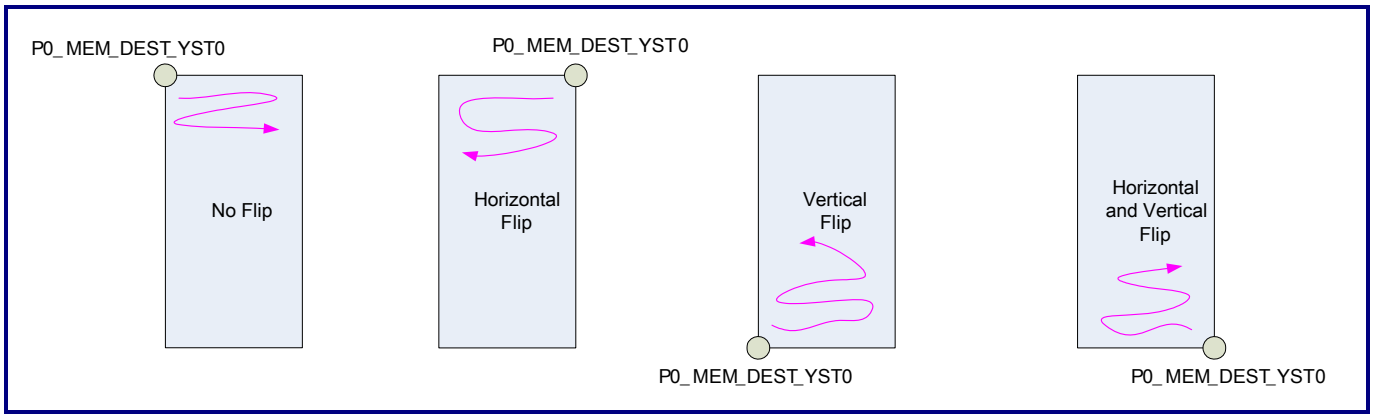



**Figure 19-7. Vertical and Horizontal Flip Address Control Example**

## 19.4.72 P1DMA0 (Offset 0x0204)

**Table 19-77. P1DMA0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | - | - | - | Reserved |
| [30:28] | P1DMA_WM | R/W | 4 | Low-level watermark of available DMA FIFO space on path 1 |
| 27 | | - | - | Reserved |
| [26:24] | P1_HRESP_DLY | R/W | 0 | AHB retry/split delay clock = 8+ rc_ahb_hresp_dly<br>For path 1 |
| [23:20] | - | - | - | Reserved |
| 19 | P1DMA_VBI_EN | R/W | 0 | VBI capture enable/disable (sync register)<br>0: Disable VBI capture<br>1: Enable VBI capture |
| [18:17] | - | - | - | Reserved |
| 16 | P1DMA_MST_EN | R/W | 0 | Multi-start address control of DMA write for Path 1<br>0: Enable frame numbers 0 start address<br>1: Enable frame numbers 0, 1 start address |
| 15 | P1DMA_MJMP | - | - | Do DMA ordering of Path 1 by jumping to the next location in distance n per n pixel for n x n macro block order. N = 8, 16<br>0: Disable<br>1: Enable |
| 14 | - | - | - | Reserved |
| [13:12] | P1DMA_ORDER | R/W | 0 | YCbCr 4:2:2/4:2:0 data organization in three memory buffers for Path 1<br>00: Sequential order<br>Others: Reserved |
| 11 | P1_DMA_V_FLIP | R/W | 0 | Up-stream DMA vertical flipping mode switch on Path 1<br>Program different destination base address in each flipping mode<br>0: No flipping in the DMA vertical direction<br>1: Do flipping in the DMA vertical direction |
| 10 | P1_DMA_H_FLIP | R/W | 0 | Up-stream DMA horizontal flipping mode switch on Path 1<br>Program different destination base address in each flipping mode<br>0: No flipping in the DMA horizontal direction<br>1: Do flipping in the DMA horizontal direction |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [9:8] | P1DMA_SPLIT | R/W | 0 | YCbCr data arrangement in memory<br>00: Write three color components into single memory block (Packed)<br>Others: Reserved |
| [7:6] | P1DMA_C_WLEN | R/W | 0 | Cb/Cr burst write length<br>00: 1 data package for one burst<br>01: 2 data packages for one burst<br>10: 4 data packages for one burst<br>11: 8 data packages for one burst |
| [5:4] | P1DMA_Y_WLEN | R/W | 0 | Y burst write length<br>00: 4 data packages for one burst<br>01: 8 data packages for one burst<br>10: 12 data packages for one burst<br>11: 16 data packages for one burst |
| [3:2] | - | - | - | Reserved |
| [1:0] | P1DMA_WLEN | R/W | 0 | DMA write burst length on path 1<br>00: 8 data packages for one burst<br>01: 16 data packages for one burst<br>10 ~ 11: Reserved |

### 19.4.73 P0DMA1 (Offset 0x0218)

**Table 19-78. PDMA1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | P0_SUB_WIN_PIT | R/W | 0 | Pitch of the destination window for path-0 DMA |

### 19.4.74 P1DMA1 (Offset 0x021C)

**Table 19-79. P1DMA1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | P1_SUB_WIN_PIT | R/W | 0 | Pitch of the destination window for path-1 DMA |

## 19.4.75 P0MDEST0 (Offset 0x0220)

**Table 19-80. P0MDEST0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P0_MEM_DEST_YST0 | R/W | 0 | Destination frame buffer 0 for Path 0 of the Y start address<br>DWORD alignment |

## 19.4.76 P0MDEST1 (Offset 0x0224)

**Table 19-81. P0MDEST1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P0_MEM_DEST_YST1 | R/W | 0 | Destination frame buffer 1 for Path 0 of the Y start address<br>DWORD alignment |

## 19.4.77 P0MDEST2 (Offset 0x0230)

**Table 19-82. P0MDEST2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P0_MEM_DEST_CBST0 | R/W | 0 | Destination frame buffer 0 for Path 0 of the Cb start address<br>DWORD alignment |

## 19.4.78 P0MDEST3 (Offset 0x0234)

**Table 19-83. P0MDEST3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P0_MEM_DEST_CBST1 | R/W | 0 | Destination frame buffer 1 for Path 0 of the Cb start address<br>DWORD alignment |

## 19.4.79 P0MDEST4 (Offset 0x0240)

**Table 19-84. P0MDEST4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P0_MEM_DEST_CRST0 | R/W | 0 | Destination frame buffer 0 for path 0 of the Cr start address<br>DWORD alignment |

## 19.4.80 P0MDEST5 (Offset 0x0244)

**Table 19-85. P0MDEST5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P0_MEM_DEST_CRST1 | R/W | 0 | Destination frame buffer 1 for path 0 of the Cr start address<br>DWORD alignment |

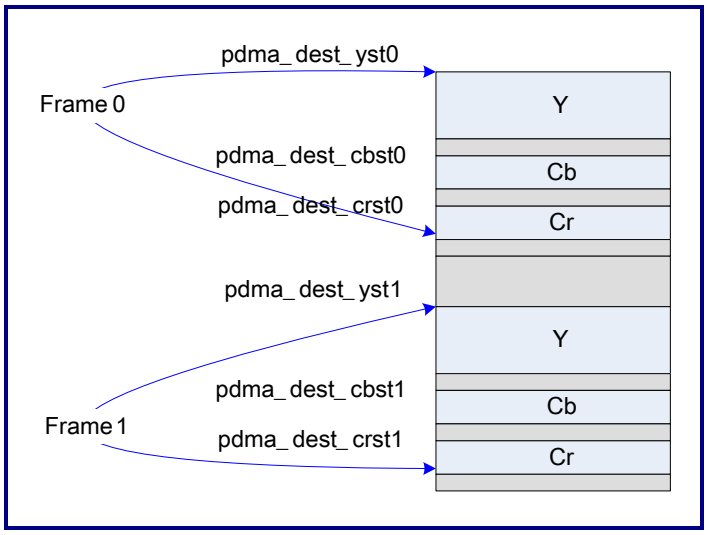



**Figure 19-8. Start Address of Destination Frame Buffer**

### 19.4.81 P1MDEST0 (Offset 0x0250)

**Table 19-86. P1MDEST0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_YST0 | R/W | 0 | Destination frame buffer 0 for path 1 of the Y start address<br>DWORD alignment |

### 19.4.82 P1MDEST1 (Offset 0x0254)

**Table 19-87. P1MDEST1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_YST1 | R/W | 0 | Destination frame buffer 1 for path 1 of the Y start address<br>DWORD alignment |

### 19.4.83 P1MDEST2 (Offset 0x0260)

**Table 19-88. P1MDEST2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_CBST0 | R/W | 0 | Destination frame buffer 0 for path 1 of the Cb start address<br>DWORD alignment |

### 19.4.84 P1MDEST3 (Offset 0x0264)

**Table 19-89. P1MDEST3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_CBST1 | R/W | 0 | Destination frame buffer 1 for path 1 of the Cb start address<br>DWORD alignment |

## 19.4.85 P1MDEST4 (Offset 0x0270)

**Table 19-90. P1MDEST4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_CRST0 | R/W | 0 | Destination frame buffer 0 for path 1 of the Cr start address<br>DWORD alignment |

## 19.4.86 P1MDEST5 (Offset 0x0274)

**Table 19-91. P1MDEST5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_CRST1 | R/W | 0 | Destination frame buffer 1 for path 1 of the Cr start address<br>DWORD alignment |

## 19.4.87 VBICTRL0 (Offset 0x0290)

**Table 19-92. VBICTRL0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_VBI_ST0 | R/W | 0 | Star address of VBI frame number 0 for path-1 DMA<br>DWORD alignment |

## 19.4.88 VBICTRL1 (Offset 0x0294)

**Table 19-93. VBICTRL1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | P1_MEM_DEST_VBI_ST1 | R/W | 0 | Start address of VBI frame number 1 for path-1 DMA<br>DWORD alignment |

## 19.4.89 VBICTRL2 (Offset 0x02A0)

**Table 19-94. VBICTRL2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:21] | - | - | - | Reserved |
| [20:16] | VBI_Y | R/W | 0 | VBI start line |
| [15:12] | - | - | - | Reserved |
| [11:0] | VBI_X | R/W | 0 | VBI start pixel |

## 19.4.90 VBICTRL3 (Offset 0x02B0)

**Table 19-95. VBICTRL3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:21] | - | - | - | Reserved |
| [20:16] | VBI_HEIGHT | R/W | 00 | Total VBI lines captured for each field |
| [15:12] | - | - | - | Reserved |
| [11:0] | VBI_WIDTH | R/W | - | Total VBI pixels captured for each line<br>In multiples of 4 |

## 19.4.91 P0OSDFONT (Offset 0x0300)

**Table 19-96. P0OSDFONT Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | P0_FONT_RAM_READ | R/W | 0 | OSD font memory read/write setting, default write |
| [30:29] | - | - | - | Reserved |
| [28:16] | P0_FONT_RAM_ADDR | R/W | 0 | OSD font address<br>CAP0 path 0 font RAM address ranging from 0 to 4031<br>CAP1 path 0 font RAM address ranging from 0 to 2015 |
| [15:12] | - | - | - | Reserved |
| [11:0] | P0_FONT_RAM_DATA | R/W | 0 | OSD font data |

## 19.4.92 P0OSDDISP (Offset 0x0304)

**Table 19-97. P0OSDDISP Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | P0_DISP_RAM_READ | R/W | 0 | Display ram read/write, default write |
| [30:22] | - | - | - | Reserved |
| [21:16] | P0_DISP_RAM_ADDR | R/W | 0 | Display RAM address.<br>CAP0 path 0 display RAM address ranging from 0 to 257<br>CAP1 path 0 display RAM address ranging from 0 to 127 |
| [15:8] | - | - | - | Reserved |
| [7:0] | P0_DISP_RAM_DATA | R/W | 0 | Set MSB[7] 1 to select RAM |

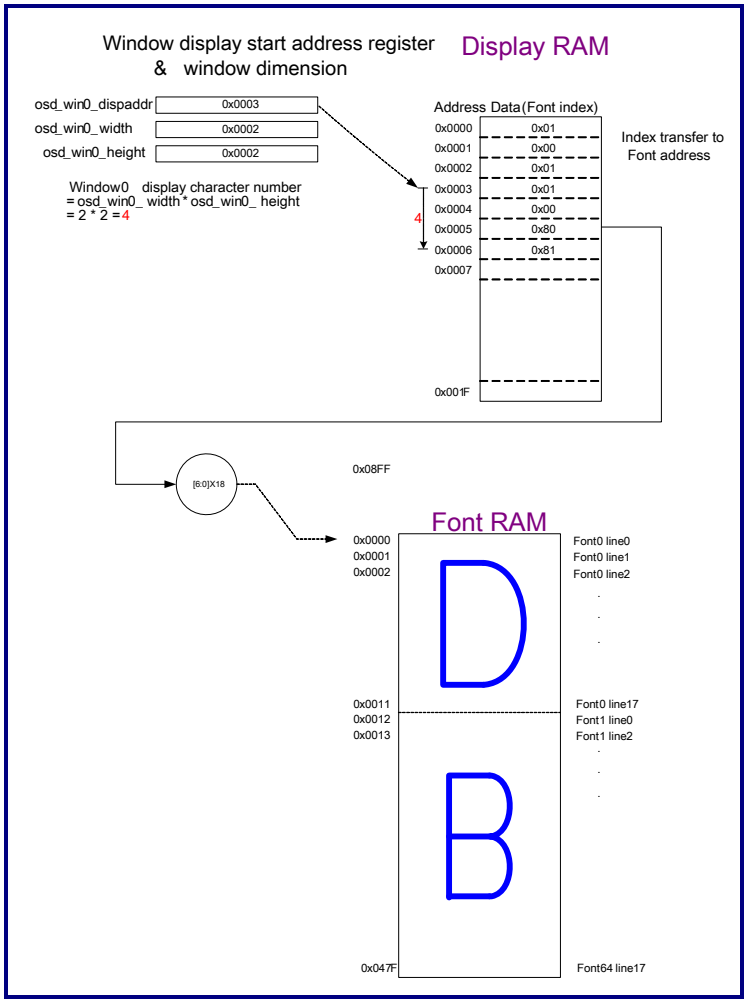



**Figure 19-9. OSD Font RAM and Display RAM**

## 19.4.93 P0OSDREAD (Offset 0x0308)

**Table 19-98. P0OSDREAD Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | P0_DISP_RAM_BUSY | R | 0 | Display RAM busy |
| 30 | P0_FONT_RAM_BUSY | R | 0 | Font memory busy |
| [29:20] | - | - | - | Reserve |
| [19:12] | P0_DISP_RAM_DATAOUT | R | 0 | Display RAM read data out |
| [11:0] | P0_FONT_RAM_DATAOUT | R | 0 | Font memory read data out |

## 19.4.94 P0OSDEN (Offset 0x030C)

**Table 19-99. P0OSDEN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:17] | - | - | - | Reserved |
| 16 | P0_OSD_EDGE_SMTH_EN | R/W | 0 | Edge smooth enable function<br>0: Disable<br>1: Enable |
| [15:12] | - | - | - | Reserved |
| 11 | P0_MSK_WIN7_EN | R/W | 0 | Mask window 7 enable (Sync. register) |
| 10 | P0_MSK_WIN6_EN | R/W | 0 | Mask window 6 enable (Sync. register) |
| 9 | P0_MSK_WIN5_EN | R/W | 0 | Mask window 5 enable (Sync. register) |
| 8 | P0_MSK_WIN4_EN | R/W | 0 | Mask window 4 enable (Sync. register) |
| 7 | P0_MSK_WIN3_EN | R/W | 0 | Mask window 3 enable (Sync. register) |
| 6 | P0_MSK_WIN2_EN | R/W | 0 | Mask window 2 enable (Sync. register) |
| 5 | P0_MSK_WIN1_EN | R/W | 0 | Mask window 1 enable (Sync. register) |
| 4 | P0_MSK_WIN0_EN | R/W | 0 | Mask window 0 enable (Sync. register) |
| 3 | P0_OSD_WIN3_EN | R/W | 0 | Window 3 enable (Sync. register) |
| 2 | P0_OSD_WIN2_EN | R/W | 0 | Window 2 enable (Sync. register) |
| 1 | P0_OSD_WIN1_EN | R/W | 0 | Window 1 enable (Sync. register) |
| 0 | P0_OSD_WIN0_EN | R/W | 0 | Window 0 enable (Sync. register) |

## 19.4.95 P0OSDPAT0 (Offset 0x0310)

**Table 19-100. P0OSDPAT0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P0_OSD_PAL0_Y | R/W | 0 | Palette color 0 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_PAL0_CB | R/W | 0 | Palette color 0 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_OSD_PAL0_CR | R/W | 0 | Palette color 0 Cr |

## 19.4.96 P0OSDPAT1 (Offset 0x0314)

**Table 19-101. P0OSDPAT1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P0_OSD_PAL1_Y | R/W | 0 | Palette color 1 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_PAL1_CB | R/W | 0 | Palette color 1 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_OSD_PAL1_CR | R/W | 0 | Palette color 1 Cr |

## 19.4.97 P0OSDPALT2 (Offset 0x0318)

**Table 19-102. P0OSDPAT2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P0_OSD_PAL2_Y | R/W | 0 | Palette color 2 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_PAL2_CB | R/W | 0 | Palette color 2 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_OSD_PAL2_CR | R/W | 0 | Palette color 2 Cr |

## 19.4.98 P0OSDPAT3 (Offset 0x031C)

**Table 19-103. P0OSDPAT3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P0_OSD_PAL3_Y | R/W | 0 | Palette color 3 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_PAL3_CB | R/W | 0 | Palette color 3 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_OSD_PAL3_CR | R/W | 0 | Palette color 3 Cr |

## 19.4.99 P0OSDPAT4 (Offset 0x0320)

**Table 19-104. P0OSDPAT4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P0_OSD_PAL4_Y | R/W | 0 | Palette color 4 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_PAL4_CB | R/W | 0 | Palette color 4 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_OSD_PAL4_CR | R/W | 0 | Palette color 4 Cr |

## 19.4.100 P0OSDPAT5 (Offset 0x0324)

**Table 19-105. P0OSDPAT5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P0_OSD_PAL5_Y | R/W | 0 | Palette color 5 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_PAL5_CB | R/W | 0 | Palette color 5 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_OSD_PAL5_CR | R/W | 0 | Palette color 5 Cr |

## 19.4.101 P0OSDPAT6 (Offset 0x0328)

**Table 19-106. P0OSDPAT6 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P0_OSD_PAL6_Y | R/W | 0 | Palette color 6 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_PAL6_CB | R/W | 0 | Palette color 6 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P0_OSD_PAL6_CR | R/W | 0 | Palette color 6 Cr |

## 19.4.102 P0OSDCOR0 (Offset 0x0330)

**Table 19-107. P0OSDCOR0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | P0_OSD_WIN0_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | P0_OSD_WIN0_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [2:0] | P0_OSD_WIN0_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

### 19.4.103 P0OSDWSZ0 (Offset 0x0334)

**Table 19-108. P0OSDWSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | P0_OSD_WIN0_DISPADDR | R/W | 0 | Window 0 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | P0_OSD_WIN0_HEIGHT | R/W | 0 | Window 0 height (Sync. register)<br>Window 0 height = (OSD_WIN0_FONT_H +OSD_WIN0_FSPACE_ROW) x OSD_WIN0_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | P0_OSD_WIN0_WIDTH | R/W | 0 | Window 0 width (Sync. register)<br>Window 0 width = (OSD_WIN0_FONT_W +OSD_WIN0_FSPACE_COL) x OSD_WIN0_WIDTH |

## 19.4.104 P0OSDSSZ0 (Offset 0x0338)

**Table 19-109. P0OSDSSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | P0_OSD_WIN0_POS_Y | R/W | 0 | Window 0 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_OSD_WIN0_POS_X | R/W | 0 | Window 0 start point X (Sync. register) |

## 19.4.105 P0OSDFSZ0 (Offset 0x033C)

**Table 19-110. P0OSDFSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [27:24] | P0_OSD_WIN0_FSPACE_ROW | R/W | 0 | Window 0 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | P0_OSD_WIN0_FSPACE_COL | R/W | 0 | Window 0 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_WIN0_FONT_H | R/W | 0x8 | Window 0 font height 7 ~ 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | P0_OSD_WIN0_FONT_W | R/W | 0xC | Window 0 font width 7 ~ 12 (Sync. register) |

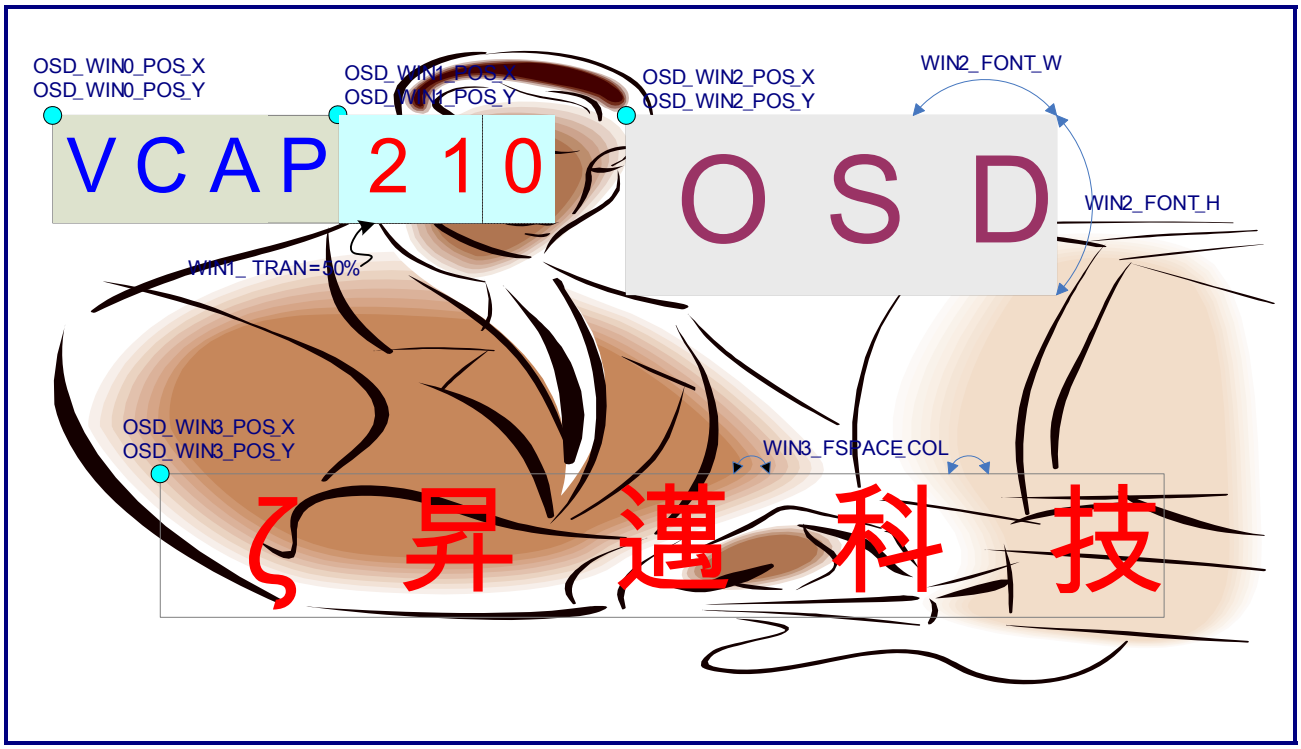



**Figure 19-10. Example of OSD**

## 19.4.106 P0OSDCOR1 (Offset 0x0340)

**Table 19-111. P0OSDCOR1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | P0_OSD_WIN1_TRAN | R/W | 0 | Window font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [6:4] | P0_OSD_WIN1_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| [2:0] | P0_OSD_WIN1_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

## 19.4.107 P0OSDWSZ1 (Offset 0x0344)

**Table 19-112. P0OSDWSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | P0_OSD_WIN1_DISPADDR | R/W | 0 | Window 1 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | P0_OSD_WIN1_HEIGHT | R/W | 0 | Window 1 height (Sync. register)<br>Window 1 height = (OSD_WIN1_FONT_H +OSD_WIN1_FSPACE_ROW) x OSD_WIN1_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | P0_OSD_WIN1_WIDTH | R/W | 0 | Window 1 width (Sync. register)<br>Window 1 width = (OSD_WIN1_FONT_W + OSD_WIN1_FSPACE_COL) x OSD_WIN1_WIDTH |

## 19.4.108 P0OSDSSZ1 (Offset 0x0348)

**Table 19-113. P0OSDSSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | P0_OSD_WIN1_POS_Y | R/W | 0 | Window 1 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_OSD_WIN1_POS_X | R/W | 0 | Window 1 start point X (Sync. register) |

## 19.4.109 P0OSDFSZ1 (Offset 0x034C)

**Table 19-114. P0OSDFSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | P0_OSD_WIN1_FSPACE_ROW | R/W | 0 | Window 1 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | P0_OSD_WIN1_FSPACE_COL | R/W | 0 | Window 1 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_WIN1_FONT_H | R/W | 0x8 | Window 1 font height ranging from 7 to 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | P0_OSD_WIN1_FONT_W | R/W | 0xC | Window 1 font width ranging from 7 to 12 (Sync. register) |

### 19.4.110 P0OSDCOR2 (Offset 0x0350)

**Table 19-115. P0OSDCOR2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | P0_OSD_WIN2_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | P0_OSD_WIN2_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| [2:0] | P0_OSD_WIN2_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

## 19.4.111 P0OSDWSZ2 (Offset 0x0354)

**Table 19-116. P0OSDWSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | P0_OSD_WIN2_DISPADDR | R/W | 0 | Window 2 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | P0_OSD_WIN2_HEIGHT | R/W | 0 | Window 2 height (Sync. register)<br>Window 2 height = (OSD_WIN2_FONT_H + OSD_WIN2_FSPACE_ROW) x OSD_WIN2_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | P0_OSD_WIN2_WIDTH | R/W | 0 | Window 2 width (Sync. register)<br>Window 2 width = (OSD_WIN2_FONT_W + OSD_WIN2_FSPACE_COL) x OSD_WIN2_WIDTH |

## 19.4.112 P0OSDSSZ2 (Offset 0x0358)

**Table 19-117. P0OSDSSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | P0_OSD_WIN2_POS_Y | R/W | 0 | Window 2 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_OSD_WIN2_POS_X | R/W | 0 | Window 2 start point X (Sync. register) |

## 19.4.113 P0OSDFSZ2 (Offset 0x035C)

**Table 19-118. P0OSDFSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:24] | P0_OSD_WIN2_FSPACE_ROW | R/W | 0 | Window 2 font row space (Sync. register) |
| [23:18] | - | - | - | Reserved |
| [17:16] | P0_OSD_WIN2_FSPACE_COL | R/W | 0 | Window 2 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_WIN2_FONT_H | R/W | 0 | Window 2 font height ranging from 7 to 12 (Sync. register) |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [7:4] | - | - | - | Reserved |
| [3:0] | P0_OSD_WIN2_FONT_W | R/W | 0 | Window 2 font width ranging from 7 to 12<br>(Sync. register) |

## 19.4.114 P0OSDCOR3 (Offset 0x0360)

**Table 19-119. P0OSDCOR3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | P0_OSD_WIN3_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | P0_OSD_WIN3_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [2:0] | P0_OSD_WIN3_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

## 19.4.115 P0OSDWSZ3 (Offset 0x0364)

**Table 19-120. P0OSDWSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | P0_OSD_WIN3_DISPADDR | R/W | 0 | Window 3 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | P0_OSD_WIN3_HEIGHT | R/W | 0 | Window 3 dimension height (Sync. register)<br>Window 3 height =<br>(OSD_WIN3_FONT_H +OSD_WIN3_FSPACE_ROW) x OSD_WIN3_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | P0_OSD_WIN3_WIDTH | R/W | 0 | Window 3 dimension width (Sync. register)<br>(OSD_WIN3_FONT_W +OSD_WIN3_FSPACE_COL) x OSD_WIN3_WIDTH |

## 19.4.116 P0OSDSSZ3 (Offset 0x0368)

**Table 19-121. P0OSDSSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | P0_OSD_WIN3_POS_Y | R/W | 0 | Window 3 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | P0_OSD_WIN3_POS_X | R/W | 0 | Window 3 start point X (Sync. register) |

## 19.4.117 P0OSDFSZ3 (Offset 0x036C)

**Table 19-122. P0OSDFSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | P0_OSD_WIN3_FSPACE_ROW | R/W | 0 | Window 3 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | P0_OSD_WIN3_FSPACE_COL | R/W | 0 | Window 3 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | P0_OSD_WIN3_FONT_H | R/W | 0x8 | Window 3 font height 7 ~ 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | P0_OSD_WIN3_FONT_W | R/W | 0xC | Window 3 font width 7 ~ 12 (Sync. register) |

## 19.4.118 P1OSDFONT (Offset 0x0370)

**Table 19-123. P1OSDFONT Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | P1_FONT_RAM_READ | R/W | 0 | OSD font memory read/write setting, default write |
| [30:29] | - | - | - | Reserved |
| [28:16] | P1_FONT_RAM_ADDR | R/W | 0 | OSD font address.<br>Both CAP0 and CAP1 path 1 font RAM address range is 0 ~ 2015. |
| [15:12] | - | - | - | Reserved |
| [11:0] | P1_FONT_RAM_DATA | R/W | 0 | OSD font data |

## 19.4.119 P1OSDDISP (Offset 0x0374)

**Table 19-124. P1OSDDISP Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | P1_DISP_RAM_READ | R/W | 0 | Display ram read/write, default write |
| [30:22] | - | - | - | Reserved |
| [21:16] | P1_DISP_RAM_ADDR | R/W | 0 | Display RAM address.<br>Both CAP0 and CAP1 path 1 display RAM address range is 0 ~ 127. |
| [15:8] | - | - | - | Reserved |
| [7:0] | P1_DISP_RAM_DATA | R/W | 0 | MSB[7], set '1' to select RAM |

## 19.4.120 P1OSDREAD (Offset 0x0378)

**Table 19-125. P1OSDREAD Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | P1_DISP_RAM_BUSY | R | 0 | Display RAM busy |
| 30 | P1_FONT_RAM_BUSY | R | 0 | Font memory busy |
| [29:20] | - | - | - | Reserve |
| [19:12] | P1_DISP_RAM_DATAOUT | R | 0 | Display RAM read data out |
| [11:0] | P1_FONT_RAM_DATAOUT | R | 0 | Font memory read data out |

## 19.4.121 P1OSDEN (Offset 0x037C)

**Table 19-126. P1OSDEN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:17] | - | - | - | Reserved |
| 16 | P1_OSD_EDGE_SMTH_EN | R/W | 0 | Edge smooth enable function<br>0: Disable<br>1: Enable |
| [15:12] | - | - | - | Reserved |
| 11 | P1_MSK_WIN7_EN | R/W | 0 | Mask Window 7 enable (Sync. register) |
| 10 | P1_MSK_WIN6_EN | R/W | 0 | Mask Window 6 enable (Sync. register) |
| 9 | P1_MSK_WIN5_EN | R/W | 0 | Mask Window 5 enable (Sync. register) |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 8 | P1_MSK_WIN4_EN | R/W | 0 | Mask Window 4 enable (Sync. register) |
| 7 | P1_MSK_WIN3_EN | R/W | 0 | Mask Window 3 enable (Sync. register) |
| 6 | P1_MSK_WIN2_EN | R/W | 0 | Mask Window 2 enable (Sync. register) |
| 5 | P1_MSK_WIN1_EN | R/W | 0 | Mask Window 1 enable (Sync. register) |
| 4 | P1_MSK_WIN0_EN | R/W | 0 | Mask Window 0 enable (Sync. register) |
| 3 | P1_OSD_WIN3_EN | R/W | 0 | Window 3 enable (Sync. register) |
| 2 | P1_OSD_WIN2_EN | R/W | 0 | Window 2 enable (Sync. register) |
| 1 | P1_OSD_WIN1_EN | R/W | 0 | Window 1 enable (Sync. register) |
| 0 | P1_OSD_WIN0_EN | R/W | 0 | Window 0 enable (Sync. register) |

### 19.4.122 P1OSDPAT0 (Offset 0x0380)

**Table 19-127. P1OSDPAT0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P1_OSD_PAL0_Y | R/W | 0 | Palette color 0 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_PAL0_CB | R/W | 0 | Palette color 0 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P1_OSD_PAL0_CR | R/W | 0 | Palette color 0 Cr |

### 19.4.123 P1OSDPAT1 (Offset 0x0384)

**Table 19-128. P1OSDPAT1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P1_OSD_PAL1_Y | R/W | 0 | Palette color 1 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_PAL1_CB | R/W | 0 | Palette color 1 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P1_OSD_PAL1_CR | R/W | 0 | Palette color 1 Cr |

## 19.4.124 P1OSDPALT2 (Offset 0x0388)

**Table 19-129. P1OSDPAT2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P1_OSD_PAL2_Y | R/W | 0 | Palette color 2 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_PAL2_CB | R/W | 0 | Palette color 2 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P1_OSD_PAL2_CR | R/W | 0 | Palette color 2 Cr |

## 19.4.125 P1OSDPAT3 (Offset 0x038C)

**Table 19-130. P1OSDPAT3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P1_OSD_PAL3_Y | R/W | 0 | Palette color 3 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_PAL3_CB | R/W | 0 | Palette color 3 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P1_OSD_PAL3_CR | R/W | 0 | Palette color 3 Cr |

## 19.4.126 P1OSDPAT4 (Offset 0x0390)

**Table 19-131. P1OSDPAT4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P1_OSD_PAL4_Y | R/W | 0 | Palette color 4 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_PAL4_CB | R/W | 0 | Palette color 4 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P1_OSD_PAL4_CR | R/W | 0 | Palette color 4 Cr |

## 19.4.127 P1OSDPAT5 (Offset 0x0394)

**Table 19-132. P1OSDPAT5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P1_OSD_PAL5_Y | R/W | 0 | Palette color 5 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_PAL5_CB | R/W | 0 | Palette color 5 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P1_OSD_PAL5_CR | R/W | 0 | Palette color 5 Cr |

## 19.4.128 P1OSDPAT6 (Offset 0x0398)

**Table 19-133. P1OSDPAT6 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | P1_OSD_PAL6_Y | R/W | 0 | Palette color 6 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_PAL6_CB | R/W | 0 | Palette color 6 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | P1_OSD_PAL6_CR | R/W | 0 | Palette color 6 Cr |

## 19.4.129 P1OSDCOR0 (Offset 0x03A0)

**Table 19-134. P1OSDCOR0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | P1_OSD_WIN0_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [6:4] | P1_OSD_WIN0_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| [2:0] | P1_OSD_WIN0_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

## 19.4.130 P1OSDWSZ0 (Offset 0x03A4)

**Table 19-135. P1OSDWSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | P1_OSD_WIN0_DISPADDR | R/W | 0 | Window0 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | P1_OSD_WIN0_HEIGHT | R/W | 0 | Window0 dimension height (Sync. register)<br>Window 0 Height =<br>(OSD_WIN0_FONT_H +OSD_WIN0_FSPACE_ROW) x OSD_WIN0_HEIGHT |
| 7 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [6:0] | P1_OSD_WIN0_WIDTH | R/W | 0 | Window0 dimension width (Sync. register)<br>Window 0 Width =<br>(OSD_WIN0_FONT_W +OSD_WIN0_FSPACE_COL) x OSD_WIN0_WIDTH |

## 19.4.131 P1OSDSSZ0 (Offset 0x03A8)

**Table 19-136. P1OSDSSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | P1_OSD_WIN0_POS_Y | R/W | 0 | Window 0 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_OSD_WIN0_POS_X | R/W | 0 | Window 0 start point X (Sync. register) |

## 19.4.132 P1OSDFSZ0 (Offset 0x03AC)

**Table 19-137. P1OSDFSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:24] | P1_OSD_WIN0_FSPACE_ROW | R/W | 0 | Window 0 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | P1_OSD_WIN0_FSPACE_COL | R/W | 0 | Window 0 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_WIN0_FONT_H | R/W | 0x8 | Window 0 font height 7 ~ 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | P1_OSD_WIN0_FONT_W | R/W | 0xC | Window 0 font width 7 ~ 12 (Sync. register) |

### 19.4.133 P1OSDCOR1 (Offset 0x03B0)

**Table 19-138. P1OSDCOR1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | P1_OSD_WIN1_TRAN | R/W | 0 | Window font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | P1_OSD_WIN1_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| [2:0] | P1_OSD_WIN1_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

## 19.4.134 P1OSDWSZ1 (Offset 0x03B4)

**Table 19-139. P1OSDWSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | P1_OSD_WIN1_DISPADDR | R/W | 0 | Window 1 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | P1_OSD_WIN1_HEIGHT | R/W | 0 | Window 1 dimension height (Sync. register)<br>Window 1 height =<br>(OSD_WIN1_FONT_H +OSD_WIN1_FSPACE_ROW) x OSD_WIN1_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | P1_OSD_WIN1_WIDTH | R/W | 0 | Window 1 dimension width (Sync. register)<br>(OSD_WIN1_FONT_W +OSD_WIN1_FSPACE_COL) x OSD_WIN1_WIDTH |

## 19.4.135 P1OSDSSZ1 (Offset 0x03B8)

**Table 19-140. P1OSDSSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | P1_OSD_WIN1_POS_Y | R/W | 0 | Window 1 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | P1_OSD_WIN1_POS_X | R/W | 0 | Window 1 start point X (Sync. register) |

## 19.4.136 P1OSDFSZ1 (Offset 0x03BC)

**Table 19-141. P1OSDFSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | P1_OSD_WIN1_FSPACE_ROW | R/W | 0 | Window 1 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | P1_OSD_WIN1_FSPACE_COL | R/W | 0 | Window 1 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [12:8] | P1_OSD_WIN1_FONT_H | R/W | 0x8 | Window 1 font height 7 ~ 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | P1_OSD_WIN1_FONT_W | R/W | 0xC | Window 1 font width 7 ~ 12 (Sync. register) |

## 19.4.137 P1OSDCOR2 (Offset 0x03C0)

**Table 19-142. P1OSDCOR2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | P1_OSD_WIN2_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | P1_OSD_WIN2_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [2:0] | P1_OSD_WIN2_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

## 19.4.138 P1OSDWSZ2 (Offset 0x03C4)

**Table 19-143. P1OSDWSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | P1_OSD_WIN2_DISPADDR | R/W | 0 | Window 2 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | P1_OSD_WIN2_HEIGHT | R/W | 0 | Window 2 dimension height (Sync. register)<br>Window 2 height =<br>(OSD_WIN2_FONT_H +OSD_WIN2_FSPACE_ROW) x OSD_WIN2_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | P1_OSD_WIN2_WIDTH | R/W | 0 | Window 2 dimension width (Sync. register)<br>(OSD_WIN2_FONT_W +OSD_WIN2_FSPACE_COL) x OSD_WIN2_WIDTH |

## 19.4.139 P1OSDSSZ2 (Offset 0x03C8)

**Table 19-144. P1OSDSSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | P1_OSD_WIN2_POS_Y | R/W | 0 | Window 2 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [10:0] | P1_OSD_WIN2_POS_X | R/W | 0 | Window 2 start point X (Sync. register) |

## 19.4.140 P1OSDFSZ2 (Offset 0x03CC)

**Table 19-145. P1OSDFSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:24] | P1_OSD_WIN2_FSPACE_ROW | R/W | 0 | Window 2 font row space (Sync. register) |
| [23:18] | - | - | - | Reserved |
| [17:16] | P1_OSD_WIN2_FSPACE_COL | R/W | 0 | Window 2 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | P1_OSD_WIN2_FONT_H | R/W | 0 | Window 2 font height 7 ~ 12 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | P1_OSD_WIN2_FONT_W | R/W | 0 | Window 2 font width 7 ~ 12 (Sync. register) |

## 19.4.141 P1OSDCOR3 (Offset 0x03D0)

**Table 19-146. P1OSDCOR3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31-10 | - | - | - | Reserved |
| 9-8 | P1_OSD_WIN3_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 6-4 | P1_OSD_WIN3_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| 2-0 | P1_OSD_WIN3_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

## 19.4.142 P1OSDWSZ3 (Offset 0x03D4)

**Table 19-147. P1OSDWSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31-23 | - | - | - | Reserved |
| 22-16 | P1_OSD_WIN3_DISPADDR | R/W | 0 | Window 3 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| 14-8 | P1_OSD_WIN3_HEIGHT | R/W | 0 | Window 3 dimension height (Sync. register)<br>Window 3 height =<br>(OSD_WIN3_FONT_H +OSD_WIN3_FSPACE_ROW) x OSD_WIN3_HEIGHT |
| 7 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 6-0 | P1_OSD_WIN3_WIDTH | R/W | 0 | Window 3 dimension width (Sync. register)<br>(OSD_WIN3_FONT_W +OSD_WIN3_FSPACE_COL) x OSD_WIN3_WIDTH |

## 19.4.143 P1OSDSSZ3 (Offset 0x03D8)

**Table 19-148. P1OSDSSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31-26 | - | - | - | Reserved |
| 26-16 | P1_OSD_WIN3_POS_Y | R/W | 0 | Window 3 start point Y (Sync. register) |
| 15-11 | - | - | - | Reserved |
| 10-0 | P1_OSD_WIN3_POS_X | R/W | 0 | Window 3 start point X (Sync. register) |

## 19.4.144 P1OSDFSZ3 (Offset 0x03DC)

**Table 19-149. P1OSDFSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31-28 | - | - | - | Reserved |
| 27-24 | P1_OSD_WIN3_FSPACE_ROW | R/W | 0 | Window 3 font row space (Sync. register) |
| 23-20 | - | - | - | Reserved |
| 19-16 | P1_OSD_WIN3_FSPACE_COL | R/W | 0 | Window 3 font column space (Sync. register) |
| 15-13 | - | - | - | Reserved |
| 12-8 | P1_OSD_WIN3_FONT_H | R/W | 0x8 | Window 3 font height 7 ~ 18 (Sync. register) |
| 7-4 | - | - | - | Reserved |
| 3-0 | P1_OSD_WIN3_FONT_W | R/W | 0xC | Window 3 font width 7 ~ 12 (Sync. register) |

### 19.4.145 INTSTS (Offset 0x03E0)

**Table 19-150. INTSTS Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | FORCE_END_STS | R | - | Force frame end when video input data less line or pixel. |
| 30 | - | - | - | Reserved |
| 29 | P1DMA_NF_STS | R | - | Path-1 DMA near full (*Write One Clear*) |
| 28 | P1DMA_OVF_ERR | R | - | Path-1 DMA write overflow error (*Write One Clear*) |
| 27 | P1_AHB_ERR | R | - | AMBA AHB receives error response on path 1 (*Write One Clear*) |
| 26 | P1_NULL_ERR | R | - | Path-1 Frame done status is not found until the start of next frame (*Write One Clear*) |
| 25 | P1_FD_QUE | R | - | Queue of field/frame capture done status on path 1. (*Write One Clear*) |
| 24 | P1_FD_STS | R | - | Frame done on path 1 (*Write One Clear*) |
| [23:21] | - | - | - | Reserved |
| 20 | P1_BOT_FD_STS | R | - | Path-1 bottom field frame done (*Write One Clear*) |
| 19 | P1_TOP_FD_STS | R | - | Path-1 top field frame done (*Write One Clear*) |
| 18 | P0_BOT_FD_STS | R | - | Path-0 bottom field frame done (*Write One Clear*) |
| 17 | P0_TOP_FD_STS | R | - | Path-0 top field frame done (*Write One Clear*) |
| [16:14] | - | - | - | Reserved |
| 13 | P1DMA_NF_STS | R | - | Path-0 DMA near full (*Write One Clear*) |
| 12 | P1DMA_OVF_ERR | R | - | Path-0 DMA write overflow error (*Write One Clear*) |
| 11 | P1_AHB_ERR | R | - | AMBA AHB receives error response on path 0 (*Write One Clear*) |
| 10 | P1_NULL_ERR | R | - | Path-0 Frame done status is not found until the start of next frame (*Write One Clear*) |
| 9 | P1_FD_QUE | R | - | Queue of field/frame capture done status on path 0 (*Write One Clear*) |
| 8 | P1_FD_STS | R | - | Path-0 frame done (*Write One Clear*) |
| [7:6] | - | - | - | Reserved |
| 5 | FM_END_STS | R | - | Status of the frame end (*Write One Clear*) |
| 4 | FM_START_STS | R | - | Status of the frame start (*Write One Clear*) |
| [3:1] | - | - | - | Reserved |
| 0 | FIELD_STS | R | - | Odd/even field status of current field (*Write One Clear*) |

### 19.4.146 INTMASK (Offset 0x03E4)

**Table 19-151. INTMASK Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | FORCE_END_MASK | R/W | 1 | Mask interrupt from FORCE_END_STS. |
| 30 | - | - | - | Reserved |
| 29 | P1DMA_NF_MASK | R/W | 1 | Mask interrupt from P1DMA_NF_STS<br>0: Retain P1DMA_NF_STS interrupt source<br>1: Mask P1DMA_NF_STS interrupt source |
| 28 | P1DMA_OVF_MASK | R/W | 0 | Mask interrupt from P1DMA_OVF_ERRr<br>0: Retain Path-1 DMA write overflow interrupt<br>1: Mask Path-1 DMA write overflow interrupt |
| 27 | P1_AHB_ERR_MASK | R/W | 0 | Mask interrupt from P1_AHB_ERR on path 1<br>0: Retain AMBA AHB error interrupt on path 1<br>1: Mask AMBA AHB error interrupt on path 1 |
| 26 | P1_NULL_ERR_MASK | R/W | 1 | Mask interrupt from P1_NULL_ERR<br>0: Retain P1_NULL_ERR interrupt source<br>1: Mask P1_NULL_ERR interrupt source |
| 25 | - | - | - | Reserved |
| 24 | P1_FD_MASK | R/W | 0 | Mask interrupt from P1_FD_STS on path 1 (P1 sync reg)<br>0: Retain P1_FD_ST interrupt on path 1<br>1: Mask P1_FD_ST interrupt on path 1 |
| [23:21] | - | - | - | Reserved |
| 20 | P1_BOT_FD_MASK | R/W | 0 | Mask interrupt from P1_BOT_FD_STS on path 1 (P1 sync reg)<br>0: Retain P1_BOT_FD_STS interrupt on path 1<br>1: Mask P1_BOT_FD_STS interrupt on path 1 |
| 19 | P1_TOP_FD_MASK | R/W | 0 | Mask interrupt from P1_TOP_FD_STS on path 1 (P1 sync reg)<br>0: Retain P1_TOP_FD_STS interrupt on path 1<br>1: Mask P1_TOP_FD_STS interrupt on path 1 |
| 18 | P0_BOT_FD_MASK | R/W | 0 | Mask interrupt from P0_BOT_FD_STS on path 0 (P0 sync reg)<br>0: Retain P0_BOT_FD_STS interrupt on path 0<br>1: Mask P0_BOT_FD_STS interrupt on path 0 |
| 17 | P0_TOP_FD_MASK | R/W | 0 | Mask interrupt from P0_TOP_FD_STS on path 0 (P0 sync reg)<br>0: Retain P0_TOP_FD_STS interrupt on path 0<br>1: Mask P0_TOP_FD_STS interrupt on path 0 |
| [16:14] | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 13 | P1DMA_NF_MASK | R/W | 1 | Mask interrupt from P1DMA_NF_STS<br>0: Retain P1DMA_NF_STS interrupt source<br>1: Mask P1DMA_NF_STS interrupt source |
| 12 | PDMA_OVF_MASK | R/W | 0 | Mask interrupt from Path-0 DMA write overflow error<br>0: Retain Path-0 DMA write overflow interrupt<br>1: Mask Path-0 DMA write overflow interrupt |
| 11 | P1_AHB_ERR_MASK | R/W | 0 | Mask interrupt from P1_AHB_ERR on path 0<br>0: Retain AMBA AHB error interrupt on path 0<br>1: Mask AMBA AHB error interrupt on path 0 |
| 10 | P1_NULL_ERR_MASK | R/W | 1 | Mask interrupt from P1_NULL_ERR<br>0: Retain P1_NULL_ERR interrupt source<br>1: Mask P1_NULL_ERR interrupt source |
| 9 | - | - | - | Reserved |
| 8 | P1_FD_MASK | R/W | 0 | Mask interrupt from P1_FD_STS on path 0 (P1 sync reg)<br>0: Retain P1_FD_STS interrupt on path 0<br>1: Mask P1_FD_STS interrupt on path 0 |
| [7:6] | - | - | - | Reserved |
| 5 | FM_END_MASK | R/W | 1 | Mask interrupt from frame end<br>0: Retain interrupt from frame end<br>1: Mask interrupt from frame end |
| 4 | FM_START_MASK | R/W | 1 | Mask interrupt from frame start<br>0: Retain interrupt from frame start<br>1: Mask interrupt from frame start |
| [3:0] | - | - | - | Reserved |

## 19.5 Function Description

### 19.5.1 Video Port

This block is used to receive video data by various types of video interfaces. It can be Y/C mix in one 8-bit/10-bit data or Y/C separated in two 8-bit/10-bit data. The H/V synchronous timing can be as the time code embedded in the video stream or be referred to the H/V reference input signals.

#### 19.5.1.1 ITU-R BT. 656

**Table 19-152. ITU-R BT. 656 EAV and SAV Sequence**

| | 8-bit Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Preamble | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Status word | 1 | F | V | H | P3 | P2 | P1 | P0 |

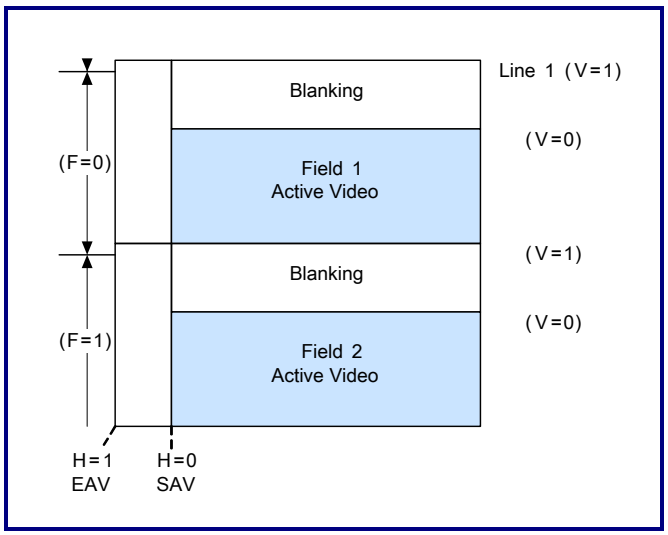



**Figure 19-11. Vertical interval of ITU-R BT. 656**

Only the SAV and EAV sequences are used to recover the video timing. Please do not make any assumption about the number of clock cycles per line or horizontal blanking interval.

### 19.5.1.2 ITU-R BT. 656 Like

| | 8-bit Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Preamble | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Status word | 1 | F | V | H | P3 | P2 | P1 | P0 |
| Vertical start (VS) | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Vertical end (VE) | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| Horizontal start (HS) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Horizontal end (HE) | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

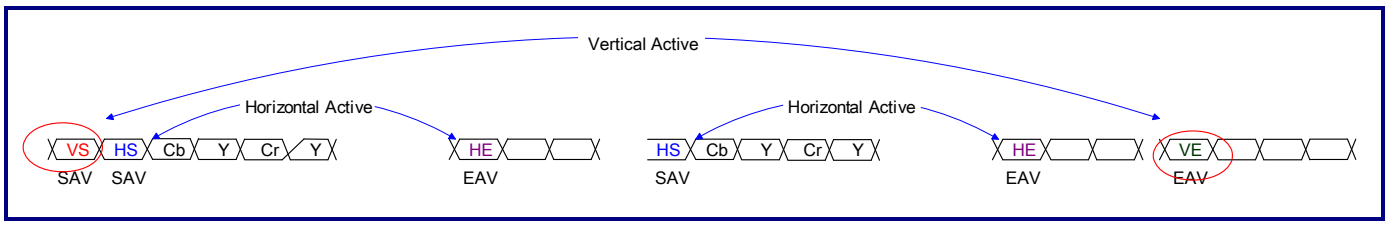



**Figure 19-12. Timing of ITU-R BT. 656 Like**

Note: There is no EAV/SAV code for the horizontal line during the vertical blank.

### 19.5.1.3 8-bit/16-bit H/V Reference Control I/F

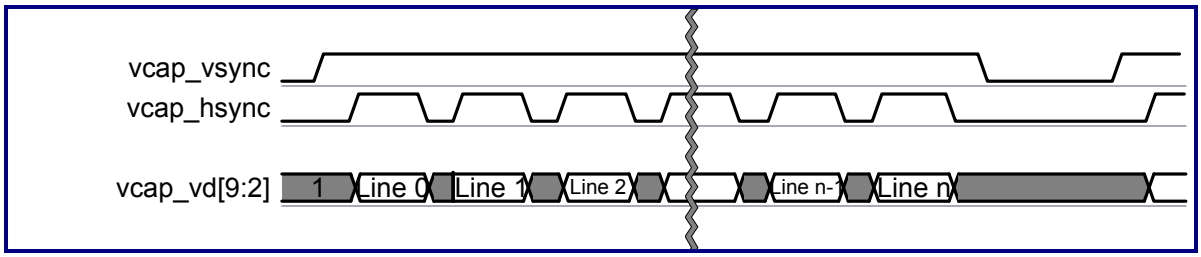



**Figure 19-13. Vertical Timing of 8-bit H/V Reference Control**

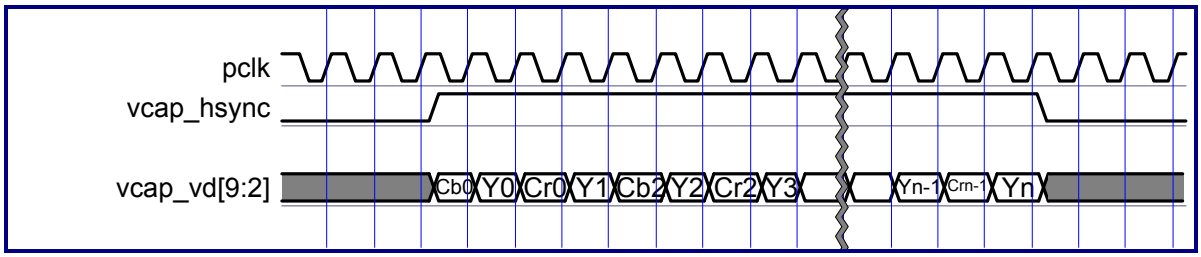



**Figure 19-14. Horizontal Timing of 8-bit H/V Reference Control**

The Sequence of Cb and Cr can be swapped.

The active polarity of vcap_hsync and vcap_vsync can be set to high or low.

### 19.5.1.4 8-bit/16-bit H/V Sync. Control I/F

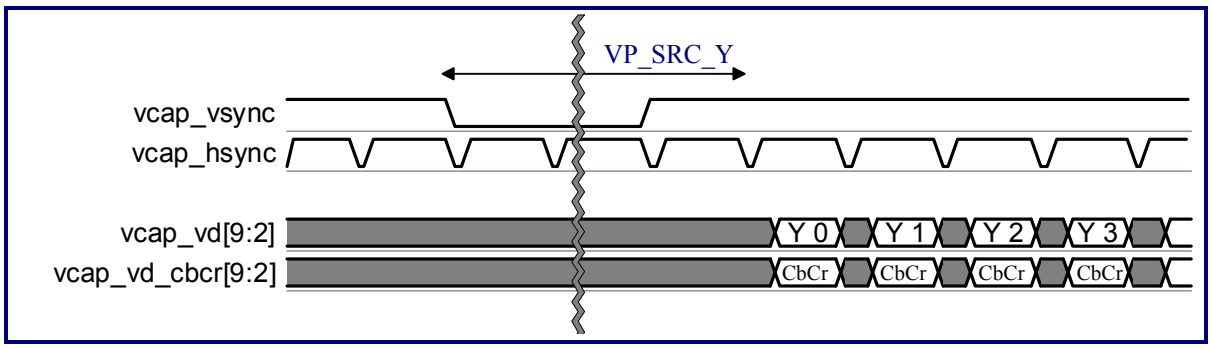



**Figure 19-15. Vertical Timing of 16-bit H/V Sync. Control**

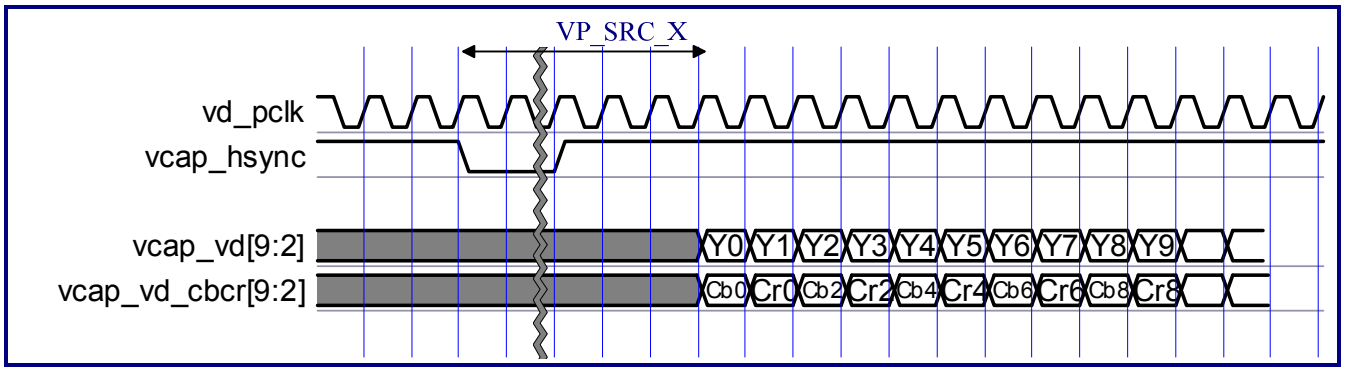



**Figure 19-16. Horizontal Timing of 16-bit H/V Sync. Control**

The sequence of Cb and Cr can be swapped.

The active polarity of vcap_hsync and vcap_vsync can be set to high or low.

### 19.5.2 De-interlacer

This block is used to perform conversion from the interlaced field to the progressive frame. The even or odd row in the original image will be removed and interpolated by the neighboring existing raw. An edge-based line average method used in the block can find the best interpolation points to generate progressive frame to reduce video artifact from interlaced field.

### 19.5.3 Size-down

This block is used to size-down the image with the horizontal and vertical ratio of the independent integer down to 1/127. The ratio of size down depends on two video capture paths. The line buffers are needed for the size-down algorithm to provide high quality size-down image

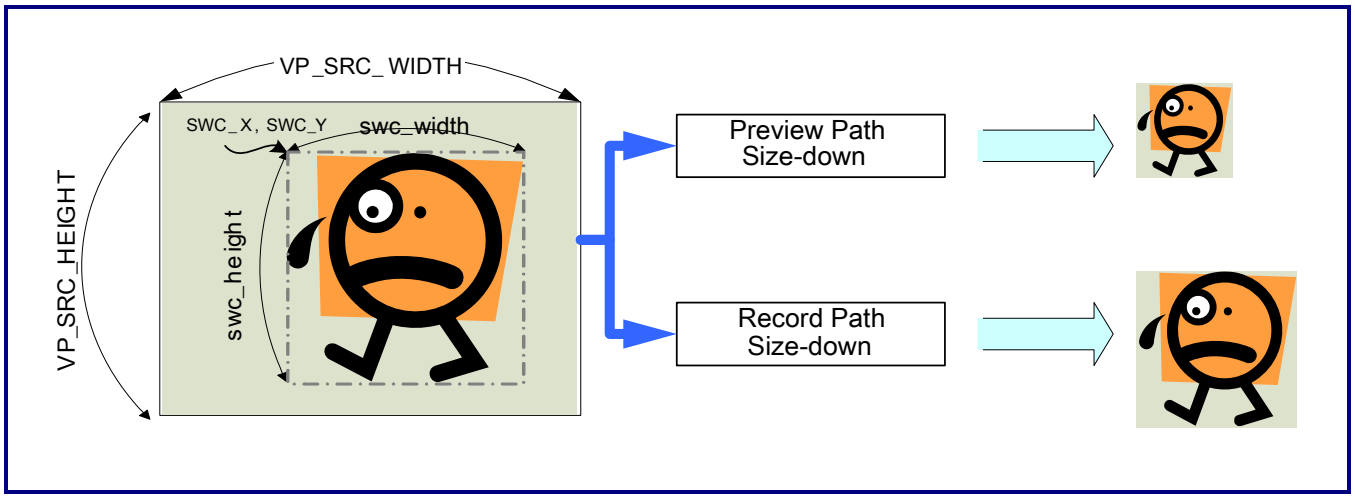



**Figure 19-17. Independent Two-path Size-down**

### 19.5.4 OSD

This block is used to paste any character to the video capture data on one path 1. Four windows with different attributes can be defined. 64 fonts with size of 12 x 18 are programmable and a maximum of 128 fonts can be shown on the display window. Adjustable font size, font space, font color, and transparency attribute are provided to work with various applications.

### 19.5.5 Color Space Conversion

This block is used to convert the color space between the YCbCr and RGB domains.

### 19.5.6 Window Clip

This block is used to clip the region of interest from image before or after sizing down. The color border with controllable width can be appended to the clipping window.

### 19.5.7 VBI Extraction

This block is used to extract the Vertical Blanking Interval (VBI) data into specified memory location

### 19.5.8 DMA

This block is used to send the video data with any color format into any memory location by scan-line or macro block order. The FIFO depth is configurable depending on the available bandwidth on the system.

### 19.5.9 YCbCr 4:2:2 Raster Order Sequence

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data | Cb0Y0Cr0 Y1 | Cb2Y2Cr2 Y3 | Cb4Y4Cr4 Y5 |

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data (CbCr swap) | Cr0Y0Cb0 Y1 | Cr2Y2Cb2 Y3 | Cr4Y4Cb4 Y5 |

# Chapter 20

# Image Sensor Processor

This chapter contains the following sections:


- 20.1 General Description
- 20.2 Features
- 20.3 Block Diagram
- 20.4 Registers and Functions
- 20.5 Block Descriptions
- 20.6 Statistics of 3A, Histogram, and Motion Detection
- 20.7 Register and Table Access

## 20.1 General Description

FTISP210 is compatible with various CMOS image sensors. It provides a sensor compensation function, making it capable of compensating the black level and lens shading phenomenon. The proprietary color interpolation algorithm implemented in this IP reconstructs the RGB components of every pixel from the Bayer raw data. The contrast enhancement, sharpness, false color suppression, and image enhancement control help users make the mages more pleasing to the human eyes. Moreover, with the consecutive size-down capability, users can resize the source image to the needed resolution. The image statistics on Auto-focus (AF), Auto-Exposure (AE), Auto-White Balance (AWB), histogram, and motion detection can be further read for 3A or other image controllers. The host processor interface is compliant with AMBA AHB 2.0, which serves as an IP core that can be integrated into the AMBA system. The raw data can also be read through AHB from other AHB slaves for the purpose of beautification, resize, or other applications.

## 20.2 Features

- Supports resolution of up to 2592x1944
- Frequency of pixel clock up to 108MHz
- Active pixel processing ability: 80 Mega pixels/sec.
- Supports 8-bit/10-bit/12-bit RGB-Bayer raw data input
- Supports optical black calibration
- Supports dark frame subtraction
- Supports lens shading compensation
- Supports auto defect pixel correction
- Supports edge-adaptive color interpolation
- Supports programmable color correction matrix
- Supports programmable RGB to YCbCr color space conversion matrix
- Supports individual RGB gamma table for gamma correction
- Supports luminance compensation curve
- Supports non-linear color saturation and hue control
- Supports contrast enhancement to lighten shadows and darken highlights
- Supports false color and chroma suppression
- Supports programmable sharpness gain and threshold control
- Supports linear seamless image resize down
- Supports image mirror, flip, and cropping

- Supports AE/AWB/AF/Motion detection statistics
- Supports Y/R/G/B histogram

## 20.3 Block Diagram

**Table 20-1. ISP Terminology**

| SI | Sensor Interface | IE | Image Enhancement |
|---|---|---|---|
| SC | Sensor Correction | YCC | YCbCr Compensation |
| CI | Color Interpolation | RS | Resizing |
| CC | Color Correction | SP | Sharpness |
| CV | Color Space Conversion | FCS | False Color Suppression |
| CS | Chroma Suppression | UDMA | Up-stream DMA |
| CE | Contrast Enhancement | DDMA | Down-stream DMA |
| HR | Hue Rotation | CU | Calculation Unit |

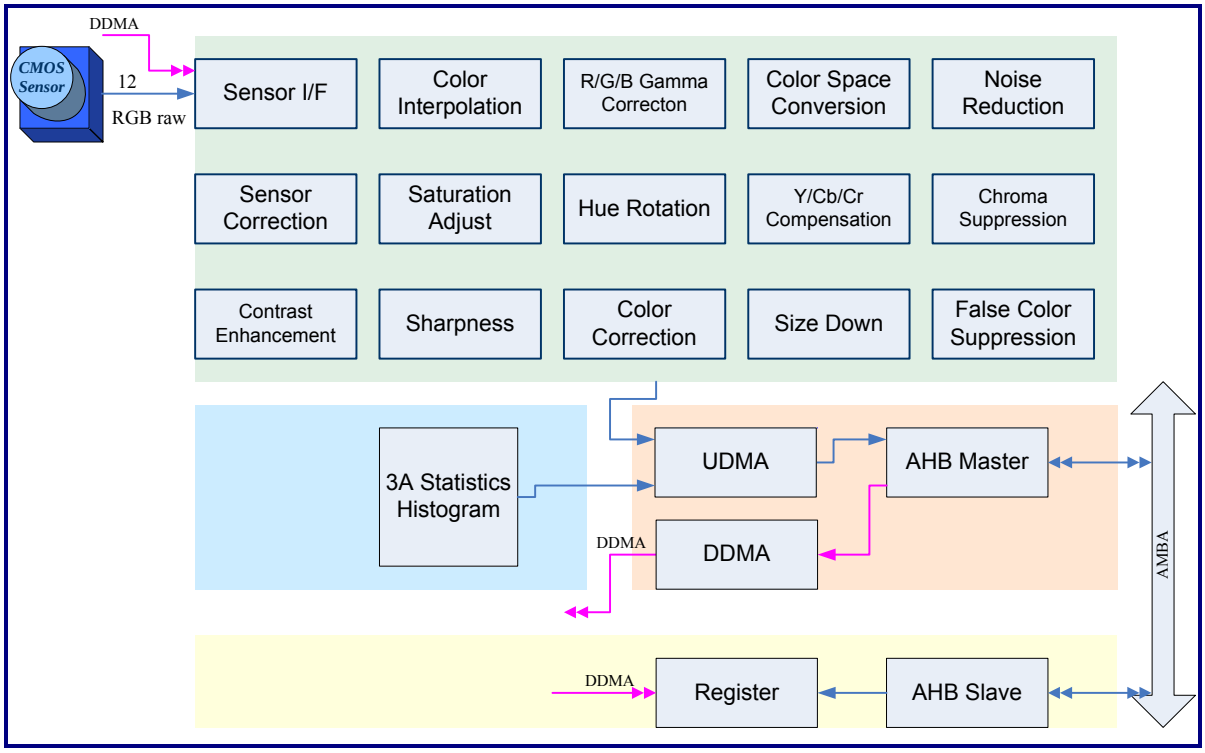



**Figure 20-1. Block Diagram**

## 20.4 Registers and Functions

### 20.4.1 Summary of Control Registers

**Table 20-2. Summary of Control Registers**

| Address | Type | Reset Value | Description |
|---|---|---|---|
| 0x0000 | R/W | 0x0800_4000 | ISP function enable |
| 0x0004 | R/W | 0 | CMI trigger and CU trigger |
| 0x0008 | R/W | 0x1000_0000 | SW Reset and Revision |
| 0x000C | R/W | 0x0001_0000 | CAP Style and Trigger |
| 0x0010 | R/W | 0 | Command buffer Control |
| 0x0014 | R/W | 0x0000_0002 | Sensor Interface |
| 0x0018 | R/W | 0 | Gamma Control |
| 0x001C | - | - | - |
| 0x0020 | R/W | 0 | Sensor Size |
| 0x0024 | R/W | 0 | Sensor Active Window Start |
| 0x0028 | R/W | 0 | Sensor Active Window Size |
| 0x002C | R/W | 0 | Sensor Blanking |
| 0x0030 | R/W | 0 | Image Target Size |
| 0x0034 | R/W | 0 | Target Crop Window Start Point |
| 0x0038 | R/W | 0 | Target Crop Window Size |
| 0x003C | R/W | 0x0000_0005 | Destination Format |
| 0x0040 | R/W | 0 | Memory Source Size |
| 0x0044 ~ 0x004B | - | - | - |
| 0x004C | R/W | 0x0000_0003 | Memory Source Format |
| 0x0050 | R/W | 0 | Optical Black Setting |
| 0x0054 | R/W | 0x3F3F_3F3F | Optical Black Threshold |
| 0x0058 | R/W | 0 | Optical Black Offset |
| 0x005C | R | 0 | Optical Black Read |
| 0x0060 | R/W | 0 | Shading Correction Center Point |
| 0x0064 ~ 0x006B | - | - | - |
| 0x006C | R/W | 0x0020_0001 | Shading Correction Shift |
| 0x0070 | R/W | 0 | Shading Correction Offset |
| 0x0074 | - | - | - |

| Address | Type | Reset Value | Description |
|---|---|---|---|
| 0x0078 | R/W | 0x0030_0010 | Raw Blending Coefficients |
| 0x007C | R/W | 0x00FF_0001 | Dark Raw Precision |
| 0x0080 | R/W | 0x0000_0001 | Defect Pixel Correction Control |
| 0x0084 | R/W | 0x4000_0000 | Defect Pixel Correction Threshold |
| 0x0088 | R/W | 0 | Crosstalk Threshold |
| 0x008C | R/W | 0x0300_0000 | Crosstalk Weighting |
| 0x0090 | R/W | 0x0400_0400 | RB Pre-gain |
| 0x0094 | R/W | 0x0400_0400 | G Pre-gain |
| 0x0098 | R/W | 0x0400_0400 | RB Gain |
| 0x009C | R/W | 0x0000_0400 | G Gain |
| 0x00A0 | R/W | 0x0000_0400 | Color Correction Matrix 00 |
| 0x00A4 | R/W | 0 | Color Correction Matrix 01 |
| 0x00A8 | R/W | 0x0400_0000 | Color Correction Matrix 10 |
| 0x00AC | R/W | 0 | Color Correction Matrix 11 |
| 0x00B0 | R/W | 0 | Color Correction Matrix 20 |
| 0x00B4 | R/W | 0x0000_0400 | Color Correction Matrix 21 |
| 0x00B8 | R/W | 0x0E54_0200 | Color Space Conversion Matrix 00 |
| 0x00BC | R/W | 0x0000_0FAD | Color Space Conversion Matrix 01 |
| 0x00C0 | R/W | 0x0259_0132 | Color Space Conversion Matrix 10 |
| 0x00C4 | R/W | 0x0000_0074 | Color Space Conversion Matrix 11 |
| 0x00C8 | R/W | 0x0EAD_0F54 | Color Space Conversion Matrix 20 |
| 0x00CC | R/W | 0x0000_0200 | Color Space Conversion Matrix 21 |
| 0x00D0 | R/W | 0x0020_0002 | Contrast Enhancement Control 0 |
| 0x00D4 | R/W | 0xFFFF_0000 | Contrast Enhancement Control 1 |
| 0x00D8 | R/W | 0x0001_0001 | Saturation Control |
| 0x00DC | R/W | 0 | Sharpness Control |
| 0x00E0 | R/W | 0x0300_1901 | FCS Control |
| 0x00E4 ~ 0x00EB | - | - | - |
| 0x00EC | R/W | 0x0100_0100 | HBLANK Extended |
| 0x00F0 ~ 0x00FF | - | - | - |
| 0x0100 | R/W | 0x0010_0010 | AE & AWB Control |
| 0x0104 | R/W | 0 | AF & MD Control |
| 0x0108 | R/W | 0 | HSTG Source |

| Address | Type | Reset Value | Description |
|---|---|---|---|
| 0x010C | - | - | - |
| 0x0110 | R/W | 0 | AE Window Start Point |
| 0x0114 | R/W | 0 | AE Window Size |
| 0x0118 | R/W | 0 | AE Window Number |
| 0x011C | R/W | 0 | AE Window Gap |
| 0x0120 | R/W | 0xFFFF_0000 | AE STA Threshold |
| 0x0124 ~ 0x013F | - | - | - |
| 0x0140 | R/W | 0 | AWB Window Start Point |
| 0x0144 | R/W | 0 | AWB Window Size |
| 0x0148 | R/W | 0 | AWB Window Number |
| 0x014C | R/W | 0 | AWB Window Gap |
| 0x0150 | R/W | 0x0000_FF00 | AWB STA Threshold |
| 0x0154 ~ 0x016F | - | - | - |
| 0x0170 | R/W | 0x8040_8040 | AF Pre-Curve Point |
| 0x0174 | R/W | 0x0100_0100 | AF Pre-Curve Slope 0 |
| 0x0178 | R/W | 0x0000_0100 | AF Pre-Curve Slope 1 |
| 0x017C | - | - | - |
| 0x0180 | R/W | 0 | AF Window Start Point |
| 0x0184 | R/W | 0 | AF Window Size |
| 0x0188 | R/W | 0 | AF Window Number |
| 0x018C | R/W | 0x0000_0004 | AF Window Gap |
| 0x0190 | R/W | 0 | AF HF1 Threshold |
| 0x0194 | R/W | 0 | AF HF2 Threshold |
| 0x0198 | R/W | 0 | AF VF Threshold |
| 0x019C | - | - | - |
| 0x01A0 | R/W | 0 | AF HF1 Coefficient 01 |
| 0x01A4 | R/W | 0 | AF HF1 Coefficient 23 |
| 0x01A8 | R/W | 0 | AF HF1 Coefficient 45 |
| 0x01AC | R/W | 0 | AF HF1 Coefficient 67 |
| 0x01B0 | R/W | 0 | AF HF2 Coefficient 01 |
| 0x01B4 | R/W | 0 | AF HF2 Coefficient 23 |
| 0x01B8 | R/W | 0 | AF HF2 Coefficient 45 |
| 0x019C | R/W | 0 | AF HF2 Coefficient 67 |

| Address | Type | Reset Value | Description |
|---|---|---|---|
| 0x01C0 | R/W | 0 | AF VF Coefficient 01 |
| 0x01C4 | R/W | 0 | AF VF Coefficient 23 |
| 0x01C8 | R/W | 0 | AF VF Coefficient 4 |
| 0x01CC | - | - | - |
| 0x01D0 | R/W | 0 | MD Window Start Point |
| 0x01D4 | R/W | 0 | MD Window Size |
| 0x01D8 | R/W | 0 | MD Window Number |
| 0x01DC | R/W | 0 | MD Window Gap |
| 0x01E0 | R/W | 0x01CC_0007 | MD Control 0 |
| 0x01E4 | R/W | 0x7FE0_0020 | MD Control 1 |
| 0x01E8 | R/W | 0x0000_0F0B | MD Control 2 |
| 0x01EC | R/W | 0x1CCB_0909 | MD Control 3 |
| 0x01F0 | R/W | 0 | HSTG Window Start Point |
| 0x01F4 | R/W | 0 | HSTG Window Size |
| 0x01F8 ~ 0x01FF | - | - | - |
| 0x0200 | R/W | 0x0000_00F5 | DMA Control |
| 0x0204 | R/W | 0x0022_0000 | DMA FIFO Water Mark |
| 0x0208 ~ 0x020F | - | - | - |
| 0x0210 | R/W | 0 | Source Memory Base Address |
| 0x0214 | - | - | - |
| 0x0218 | R/W | 0 | Source Memory Pitch |
| 0x021C | - | - | - |
| 0x0220 | R/W | 0 | Destination Memory Base Address 0 |
| 0x0224 | R/W | 0 | Destination Memory Base Address 1 |
| 0x0228 | R/W | 0 | Destination Memory Pitch |
| 0x022C | - | - | - |
| 0x0230 | R/W | 0 | Destination Memory Cb Base Address 0 |
| 0x0234 | R/W | 0 | Destination Memory Cb Base Address 1 |
| 0x0238 ~ 0x023F | - | - | - |
| 0x0240 | R/W | 0 | Destination Memory Cr Base Address 0 |
| 0x0244 | R/W | 0 | Destination Memory Cr Base Address 1 |
| 0x0248 ~ 0x024F | - | - | - |
| 0x0280 | R/W | 0 | Command 0 Base Address |

| Address | Type | Reset Value | Description |
|---|---|---|---|
| 0x0284 | R/W | 0 | Command X Base Address |
| 0x0288 | R/W | 0 | Command 1 Base Address |
| 0x028C ~ 0x028F | - | - | - |
| 0x0290 | R/W | 0 | AE Base Address 0 |
| 0x0294 | R/W | 0 | AE Base Address 1 |
| 0x0298 | R/W | 0 | AWB Base Address 0 |
| 0x029C | R/W | 0 | AWB Base Address 1 |
| 0x02A0 | R/W | 0 | AF Base Address 0 |
| 0x02A4 | R/W | 0 | AF Base Address 1 |
| 0x02A8 | R/W | 0 | HSTG Base Address |
| 0x02AC | R/W | 0 | MD Base Address |
| 0x02B0 ~ 0x2FF | - | - | - |
| 0x0300 | R/W | 0 | Line trigger threshold |
| 0x03D0 | R | 0 | Sensor I/F Image Size Info |
| 0x03D4 | R | 0 | Size down Image Size Info |
| 0x03D8 | R | 0 | Frame Counter |
| 0x03DC | R/W | 0 | User Defined Register |
| 0x03E0 | R | 0 | STA Ready Status |
| 0x03E4 | R | 0 | ISP Status |
| 0x03E8 | R | 0 | ISP Interrupt |
| 0x03EC | R/W | 0x000F_0000 | ISP Interrupt Mask |
| 0x03F0 | R | 0 | ISP State Machine 0 |
| 0x03F4 | R | 0 | ISP State Machine 1 |
| 0x03F8 | R/W | 0 | ISP Test Pattern |
| 0x03FC | - | - | - |
| 0x0400 | R/W | 0 | RDN Filter Threshold |
| 0x0404 | R/W | 0 | RDN Filter Weighting |
| 0x0408 | R/W | 0 | CDN Filter Threshold |
| 0x040C | R/W | 0 | CDN Filter Weighting |
| 0x0410 | R/W | 0 | SP LPF Control |
| 0x0414 ~ 0x043F | - | - | - |
| 0x0440 | R/W | 0x0400_0400 | SHDC RB Up Kx |
| 0x0444 | R/W | 0x0400_0400 | SHDC G Up Kx |

| Address | Type | Reset Value | Description |
|---|---|---|---|
| 0x0448 | R/W | 0x0400_0400 | SHDC RB Down Kx |
| 0x044C | R/W | 0x0400_0400 | SHDC G Down Kx |
| 0x0450 | R/W | 0x0400_0400 | SHDC RB Right Kx |
| 0x0454 | R/W | 0x0400_0400 | SHDC G Right Kx |
| 0x0458 | R/W | 0x0400_0400 | SHDC RB Left Kx |
| 0x045C | R/W | 0x0400_0400 | SHDC GLeft Kx |
| 0x0460 ~ 0x07FF | - | - | - |
| 0x0800 ~ 0x0810 | R/W | - | SP Gain |
| 0x0820 ~ 0x083C | R/W | - | SP Weighting |
| 011000 ~ 0x25FC | R/W | - | Gamma R LUT |
| 0x3000 ~ 0x45FC | R/W | - | Gamma G LUT |
| 0x5000 ~ 0x65FC | R/W | - | Gamma B LUT |
| 0x7000 ~ 0x7FFC | R/W | - | Raw Mapping Curve |
| 0x2800 ~ 0x29FC | R/W | - | YCC Luma Curve |
| 0x2C00 ~ 0x2CFC | R/W | - | YCC Chroma Curve |
| 0x4800 ~ 0x48FC | R/W | - | Chroma Suppression Gain |
| 0x4900 ~ 0x493C | R/W | - | Chroma Suppression Weighting |
| 0x4A00 ~ 0x4A7C | R/W | - | IE Saturation Gain |
| 0x4B00 ~ 0x4B58 | R/W | - | IE Hue Ratio |
| 0x4C00 ~ 0x4C3F | R/W | - | I IE Saturation to Y Offset |
| 0x4D00 ~ 0x4D58 | R/W | - | Hue rotation $\theta$ table |
| 0x4E00 ~ 0x4E1C | R/W | - | CE fine tuning Table |
| 0x4F00 ~ 0x4F7C | R/W | - | Back Chroma Suppression Gain |

## 20.4.2 ISP Global Registers

### 20.4.2.1 ISP Function Enable Register (Offset: 0x0000)

**Table 20-3. ISP Function Enable Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | CAPTURE_EN | R/W | 0 | Capture sensor data enable switch<br>ISP HW will launch CU if this bit changed from '0' to '1'.<br>0: Disable<br>1: Enable |
| [30:29] | - | - | - | Reserved |
| 28 | ALBUM_EN | R/W | 0 | Photo album capture mode switch<br>0: Disable<br>1: Enable |
| 27 | FD_SIG_EN | R/W | 1 | isp_fm_done output signal enable switch<br>0: Disable<br>1: Enable |
| 26 | STOP_SIG_EN | R/W | 0 | External signal to stop the ISP capture image data<br>0: Normal operation<br>1: Enable the external stop signal |
| [25:23] | - | - | | Reserved |
| 22 | SDN_EN | R/W | 0 | SP de-noise function switch<br>0: Disable<br>1: Enable |
| 21 | CDN_EN | R/W | 0 | CI de-noise function switch<br>0: Disable<br>1: Enable |
| 20 | RDN_EN | R/W | 0 | Raw de-noise function switch<br>0: Disable<br>1: Enable |
| 19 | STA_MD_EN | R/W | 0 | Motion detection statistics switch<br>0: Disable<br>1: Enable |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 18 | STA_HSTG_EN | R/W | 0 | Histogram statistics switch<br>0: Disable<br>1: Enable |
| 17 | STA_AF_EN | R/W | 0 | AF statistics switch<br>0: Disable<br>1: Enable |
| 16 | STA_AEWB_EN | R/W | 0 | AE/AWB statistics switch<br>0: Disable<br>1: Enable |
| 15 | DFS_EN | R/W | 0 | Dark frame subtraction function switch<br>0: Disable<br>1: Enable |
| 14 | DTO_EN | R/W | 1 | Image data to up-stream DMA switch<br>0: Disable<br>1: Enable |
| 13 | BCS_EN | R/W | 0 | Back Chroma Suppression function switch<br>0: Disable<br>1: Enable |
| 12 | HR_EN | R/W | 0 | Hue rotation function switch<br>0: Disable<br>1: Enable |
| 11 | SP_EN | R/W | 0 | Sharpness function switch<br>0: Disable<br>1: Enable |
| 10 | CE_EN | R/W | 1 | Contrast Enhancement function switch<br>0: Disable<br>1: Enable |
| 9 | FCS_EN | R/W | 0 | False Color Suppression function switch<br>0: Disable<br>1: Enable |
| 8 | IE_EN | R/W | 0 | Image Enhancement function switch<br>0: Disable<br>1: Enable |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 7 | CS_EN | R/W | 0 | Chroma Suppression function switch<br>0: Disable<br>1: Enable |
| 6 | YCC_EN | R/W | 0 | YCbCr Compensation function switch<br>0: Disable<br>1: Enable |
| 5 | GM_EN | R/W | 0 | RGB Gamma function switch<br>0: Disable<br>1: Enable |
| 4 | CTK_EN | R/W | 0 | Gb/Gr crosstalk removal function switch<br>0: Disable<br>1: Enable |
| 3 | DPC_EN | R/W | 0 | Defect Pixel Correction function switch<br>0: Disable<br>1: Enable |
| 2 | ABLC_EN | R/W | 0 | Automatic optical black level calibration switch<br>0: Disable<br>1: Enable |
| 1 | RMC_EN | R/W | 0 | Raw-mapping Curve function switch<br>0: Disable<br>1: Enable |
| 0 | SHDC_EN | R/W | 0 | Lens shading compensation function switch<br>0: Disable<br>1: Enable |

### 20.4.2.2 CMI and CU Register (Offset: 0x0004)

**Table 20-4. CMI and CU Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | FORCE_CU | R/W | 0 | Force CU to perform calculation (Auto clear)<br>0: Normal operation<br>1: Force CU to perform calculation |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [30:29] | - | - | - | Reserved |
| 28 | CU_TYPE | R/W | 1 | CU calculation type select<br>0: Partial CU calculation. CU will skip the statistics related operation.<br>1: Full CU calculation |
| [27:17] | - | - | - | Reserved |
| 16 | SCBUFX_UPD | R/W | 0 | Loading command buffer X to registers after the end of frame done<br>The priority of CMD buffer X is higher than other CMD buffers.<br>This bit will be cleared automatically until the end of updating (Auto clear).<br>0: Normal operation<br>1: Trigger the command interpreter to load commands from the command buffer x after the end of frame done. |
| [15:13] | - | - | - | Reserved |
| 12 | FCBUF1_UPD | R/W | 0 | Loading command buffer 1 to registers immediately (Auto clear)<br>0: Normal operation<br>1: Trigger the command interpreter to load commands from the command buffer 1 immediately |
| [11:9] | - | - | - | Reserved |
| 8 | SCBUF1_UPD | R/W | 0 | Loading command buffer 1 to registers after the end of frame done.<br>This bit will be cleared automatically until the end of updating (Auto clear).<br>0: Normal operation<br>1: Trigger the command interpreter to load commands from the command buffer 1 after the end of frame done. |
| [7:5] | - | - | - | Reserved |
| 4 | FCBUF0_UPD | R/W | 0 | Loading command buffer 0 to registers immediately (Auto clear)<br>0: Normal operation<br>1: Trigger the command interpreter to load commands from the command buffer 0 immediately |
| [3:1] | - | - | - | Reserved |
| 0 | SCBUF0_UPD | R/W | 0 | Loading command buffer 0 to registers after the end of frame done.<br>This bit will be cleared automatically until the end of updating (Auto clear).<br>0: Normal operation<br>1: Trigger the command interpreter to load commands from the command buffer 0 after the end of frame done. |

### 20.4.2.3 SW Reset Register (Offset: 0x0008)

**Table 20-5. SW Reset Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| 28 | SW_RST | R/W | 0 | Software reset (Auto clear)<br>0: Normal operation<br>1: Reset |
| [27:24] | SW_RST_DLY | R/W | 0xF | Software resets the delay counter. |
| [23:10] | - | - | - | Reserved |
| 9 | INT_LN_CLR | R/W | 0 | Clear the STA interrupt at the end of every line automatically<br>0: User clear<br>1: Auto clear |
| 8 | INT_SYNC_CLR | R/W | 0 | Clear the interrupt source at every frame start automatically<br>0: User clear<br>1: Auto clear |
| [7:0] | REVISION | R | - | [7:4]: Major revision<br>[3:0]: Minor revision |

### 20.4.2.4 CAP Style and Trigger Register (Offset: 0x000C)

**Table 20-6. CAP Style and Trigger Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:20] | ASPECT_MJ | R/W | 0 | Resizing aspect ratio definition<br>0: Resize the output image in X-, Y-direction independent mode<br>1: Resize the output image to keep aspect ratio by X-direction major<br>2: Resize the output image to keep aspect ratio by Y-direction major<br>3: Reserved |
| [19:17] | - | - | - | Reserved |
| 16 | CENTER_XY | R/W | 1 | Behavior of sensor active window<br>0: User-specified active window start location<br>1: Active window located at the center of image<br>The start point of sensor_act_x and sensor_act_y is not concerned. |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [15:9] | - | - | - | Reserved |
| 8 | STEP_CAP_FIRE | R/W | 0 | Stepping-frame capture fire (Auto clear)<br>0: Normal operation<br>1: Fire stepping capture to process one frame |
| [7:6] | - | - | - | Reserved |
| [5:4] | CAP_STYLE | R/W | 0 | Sensor data capture style<br>0: Continuous frame capture<br>1: Stepping frame capture<br>2: CMI implicit frame capture. Don't care UDMA overflow condition<br>3: CMI explicit frame capture. Care UDMA overflow condition |
| [3:1] | - | - | - | Reserved |
| 0 | IMG_SRC_SEL | R/W | 0 | Source image selection (hclk domain)<br>0: Source image from sensor<br>1: Source image from memory |

### 20.4.2.5 Command Buffer Control Register (Offset: 0x0010)

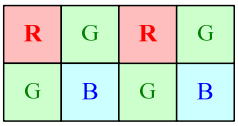
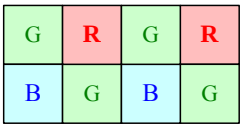
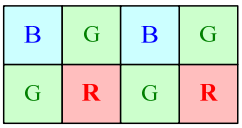
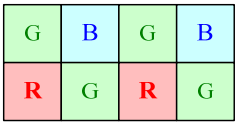
**Table 20-7. Command Buffer Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:17] | - | - | - | Reserved |
| 16 | SYNC_CBUFX_KP | R/W | 0 | Keep and repeat the action of SYNC_CBUFX_UPD<br>0: Normal operation<br>1: SYNC_CBUFX_UPD will be kept to the end of next frame |
| [15:9] | - | - | - | Reserved |
| 8 | CMI_CAP_EXTD | R/W | 0 | In the "CMI frame capture" mode, this bit determines whether or not the last CMI table related image frame will be captured.<br>0: Image capture will be stop after the final loading of CMI table<br>1: The last CMI table related image frame will be captured. |
| [7:1] | - | - | - | Reserved |
| 0 | SYNC_CBUF0_KP | R/W | 0 | Keep and repeat the action of SYNC_CBUF0_UPD.<br>0: Normal operation<br>1: SYNC_CBUF0_UPD will be kept to the end of next frame. |

### 20.4.2.6 Sensor Interface Register (Offset: 0x0014)

**Table 20-8. Sensor Interface Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| 11 | SVREF_POL | R/W | 1 | Polarity of sensor vertical control timing<br>0: Vertical control timing is active low.<br>1: Vertical control timing is active high. |
| 10 | SHREF_POL | R/W | 1 | Polarity of sensor horizontal control timing<br>0: Horizontal control timing is active low.<br>1: Horizontal control timing is active high. |
| 9 | SVREF_DEF | R/W | 0 | The definition of sensor vertical control timing<br>0: Sensor vertical control timing is treated as the Valid signal.<br>1: Sensor vertical control timing is treated as the Synchronization signal. |
| 8 | SHREF_DEF | R/W | 0 | The definition of sensor horizontal control timing<br>0: Sensor horizontal control timing is treated as the Vali signal.<br>1: Sensor horizontal control timing is treated as the Synchronization signal. |
| [7:6] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [5:4] | BAYER_TYPE | R/W | 0 | The type of sensor Bayer raw data<br>0: RG raw data sequence<br>  <br>1: GR raw data sequence<br>  <br>2: BG raw data sequence<br>  <br>3: GB raw data sequence<br>  |
| [3:2] | - | - | - | Reserved |
| [1:0] | RAW_PIX_DEPTH | R/W | 2 | Per-pixel raw data depth<br>0: 8-bit RGB Bayer raw data<br>1: 10-bit RGB Bayer raw data<br>2: 12-bit RGB Bayer raw data<br>3: Reserved |

### 20.4.2.7 Gamma Control Register (Offset: 0x0018)

**Table 20-9. Gamma Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:1] | - | - | | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 0 | GM_LUTX1 | R/W | 0 | GM LUT mapping mode<br>The function will be valid when GM LUTs are programmed by using the CMI method.<br>0: GM RGB pixel refers to three separated RGB-LUTs.<br>1: GM RGB pixel refers to one R-LUT. |

### 20.4.2.8 Sensor Size Register (Offset: 0x0020)

**Table 20-10. Sensor Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | SENSOR_HEIGHT | R/W | 0 | Height of the image sensor |
| [15:12] | - | - | - | Reserved |
| [11:0] | SENSOR_WIDTH | R/W | 0 | Width of the image sensor |

### 20.4.2.9 Sensor Active Window Start Point Register (Offset: 0x0024)

**Table 20-11. Sensor Active Window start Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | SENSOR_ACT_Y | R/W | 0 | Vertical start line of the sensor active window<br>In multiples of 2 |
| [15:12] | - | - | - | Reserved |
| [11: 0] | SENSOR_ACT_X | R/W | 0 | Horizontal start point of the sensor active window<br>In multiples of 2 |

### 20.4.2.10 Sensor Active Window Size Register (Offset: 0x0028)

**Table 20-12. Sensor Active Window Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | SENSOR_ACT_H | R/W | 0 | Height of the image sensor active window<br>In multiples of 2 |
| [15:12] | - | - | - | Reserved |
| [11:0] | SENSOR_ACT_W | R/W | 0 | Width of the image sensor active window<br>In multiples of 4 |

### 20.4.2.11 Sensor Blanking Register (Offset: 0x002C)

**Table 20-13. Sensor Blanking Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:16] | SI_HBLK_LN | R/W | 0 | Number of sensor vertical blanking lines |
| [15:10] | - | - | - | Reserved |
| [9:0] | SI_HBLK_PX | R/W | 0 | Number of sensor horizontal blanking pixels |

### 20.4.2.12 Image Target Size Register (Offset: 0x0030)

**Table 20-14. Image Target Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | TARGET_HEIGHT | R/W | 0 | Height of the output target image |
| [15:12] | - | - | - | Reserved |
| [11:0] | TARGET_WIDTH | R/W | 0 | Width of the output target image in multiples of 4 |

### 20.4.2.13 Target Crop Window Start Point Register (Offset: 0x0034)

**Table 20-15. Target Crop Window Start Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CROP_Y | R/W | 0 | Vertical-line start point of the output cropping window |
| [15:12] | - | - | - | Reserved |
| [11:0] | CROP_X | R/W | 0 | Horizontal pixel start point of the output cropping windov |

### 20.4.2.14 Target Crop Window Size Register (Offset: 0x0038)

**Table 20-16. Target Crop Window Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CROP_HEIGHT | R/W | 0 | Height of the output cropping image in multiples of 2 |
| [15:12] | - | - | - | Reserved |
| [11:0] | CROP_WIDTH | R/W | 0 | Width of the output cropping image in multiples of 4 |

### 20.4.2.15 Destination Format Register (Offset: 0x003C)

**Table 20-17. Destination Format Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:6] | - | - | - | Reserved |
| 5 | DEST_YC_SWAP | R/W | 0 | Destination of Y/CrCb sequence swap<br>Valid packed format<br>0: CbCr sequence is ahead of Y (CbYCr or CrYCb).<br>1: Y sequence is ahead of CbCr (YCbCr or YCrCb). |
| 4 | DEST_RB_SWAP | R/W | 0 | Destination of Cr/Cb sequence swap<br>Valid packed format<br>0: Cr sequence is ahead of Cb.<br>1: Cb sequence is ahead of Cr. |
| 3 | DEST_YCRANGE | R/W | 0 | 0: YCbCr is ranging from 0 to 255.<br>1: Y is ranging from 16 to 235.<br>CbCr is ranging from 16 to 240 |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [2:0] | DEST_FORMAT | R/W | 5 | Output data format<br>0, 1, 6, 7: Reserved<br>2: 8-bit Raw RGB<br>Output data are directly from MSB of 8-bit Sensor I/F.<br>3: 12-bit Raw RGB<br>Output data are directly from Sensor I/F.<br>4: Dark frame RAW<br>5: YCbCr 4:2:2 |

| Offset | 0x0~0x3 | 0x4~0x7 | 0x8~0xB |
|---|---|---|---|
| Data | Cb0Y0Cr0 Y1 | Cb2Y2Cr2 Y3 | Cb4Y4Cr4 Y5 |

| Offset | 0x0~0x3 | 0x4~0x7 | 0x8~0xB |
|---|---|---|---|
| Data (CbCr swap) | Cr0Y0Cb0 Y1 | Cr2Y2Cb2 Y3 | Cr4Y4Cb4 Y5 |

### 20.4.2.16 Memory Source Size Register (Offset: 0x0040)

**Table 20-18. Memory Source Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | MEM_SRC_H | R/W | 0 | Height of the memory source image |
| [15:12] | - | - | - | Reserved |
| [11: 0] | MEM_SRC_W | R/W | 0 | Width of the memory source image |

### 20.4.2.17 Memory Source Format Register (Offset: 0x004C)

**Table 20-19. Memory Source Format Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:3] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [2:0] | MEM_SRC_FORMAT | R/W | 3 | Data input from the main memory format<br>0, 1, 4 ~ 7: Reserved<br>2: 8-bit Raw RGB<br>Data input before the Sensor I/F<br>3: 12-bit Raw RGB<br>Data input before the Sensor I/F |

### 20.4.2.18 Optical Black Setting Register (Offset: 0x0050)

**Table 20-20. Optical Black Setting Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:18] | - | - | - | Reserved |
| [17:16] | BLC_PBIT | R/W | 0 | Precision optical black level calibration<br>0: Raw data [7:0]<br>1: Raw data [9:2]<br>2: Raw data [11:4]<br>3: Reserved |
| [15:7] | - | - | - | Reserved |
| [6:4] | BLC_LINE | R/W | 0 | Number of optical black lines |
| 3 | - | - | - | - |
| [2:0] | BLC_Y | R/W | 0 | Line offset of the optical black area |

### 20.4.2.19 Optical Black Threshold Register (Offset: 0x0054)

**Table 20-21. Optical Black Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | - | - | - | Reserved |
| [30:24] | BG_OB_H_TH | R/W | 0x3F | High threshold for the optical black level calibration at the Gb cell<br>Value below the threshold is qualified as the black level.<br>Format: 7. (8-bit raw data precision) |
| 23 | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [22:16] | RG_OB_H_TH | R/W | 0x3F | High threshold for the optical black level calibration at the Gr cell<br>Value below the threshold is qualified as the black level.<br>Format: 7. (8-bit raw data precision) |
| 15 | - | - | - | Reserved |
| [14:8] | B_OB_H_TH | R/W | 0x3F | High threshold for the optical black level calibration at the B cell<br>Value below the threshold is qualified as the black level.<br>Format: 7. (8-bit raw data precision) |
| 7 | - | - | - | Reserved |
| [6:0] | R_OB_H_TH | R/W | 0x3F | High threshold for the optical black level calibration at the R cell<br>Value below the threshold is qualified as the black level.<br>Format: 7. (8-bit raw data precision) |

### 20.4.2.20 Optical Black Offset Register (Offset: 0x0058)

**Table 20-22. Optical Black Offset Registers**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | BG_OB_OFFSET | R/W | 0 | User-specified optical black offset at the Gb cell<br>It will be valid if the auto-black level calibration is disabled.<br>Format: 8. |
| [23:16] | RG_OB_OFFSET | R/W | 0 | User-specified optical black offset at the Gr cell<br>It will be valid if the auto-black level calibration is disabled.<br>Format: 8. |
| [15:8] | B_OB_OFFSET | R/W | 0 | User-specified optical black offset at B cell.<br>It will be valid if the auto-black level calibration is disabled.<br>Format: 8. |
| [7:0] | R_OB_OFFSET | R/W | 0 | User-specified optical black offset at R cell.<br>It will be valid if the auto-black level calibration is disabled.<br>Format: 8. |

### 20.4.2.21 Optical Black Read Register (Offset: 0x005C)

**Table 20-23. Optical Black Read Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | BG_OB_DATA | R | - | Calculated result of the optical black offset at the Gb cell<br>Format: 8. |
| [23:16] | RG_OB_ DATA | R | - | Calculated result of the optical black offset at the Gr cell<br>Format: 8. |
| [15:8] | B_OB_ DATA | R | - | Calculated result of the optical black offset at the B cell<br>Format: 8. |
| [7:0] | R_OB_ DATA | R | - | Calculated result of the optical black offset at the R cell<br>Format: 8. |

### 20.4.2.22 Shading Correction Center Point Register (Offset: 0x0060)

**Table 20-24. Shading Correction Center Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [16:16] | SHDC_CENTER_Y | R/W | 0 | Y-coordinate of the lens center position |
| [15:11] | - | - | - | Reserved |
| [10:0] | SHDC_CENTER_X | R/W | 0 | X-coordinate of the lens center position |

### 20.4.2.23 Shading Correction Shift Register (Offset: 0x006C)

**Table 20-25. Shading Correction Shift Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:4] | - | - | - | Reserved |
| [3:0] | SHDC_SHIFT | R/W | 1 | Shift value of the radius square<br>0: No shift<br>1: Shift 1 bit to the right<br>2: Shift 2 bits to the right<br>~<br>15: Shift15 bits to the right |

### 20.4.2.24 Shading Correction Offset Register (Offset: 0x0070)

**Table 20-26. Shading Correction Offset Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:21] | - | - | - | Reserved |
| [20:0] | SHDC_OFFSET | R/W | 0 | Smooth shading zone |

### 20.4.2.25 Raw Blending Coefficients Register (Offset: 0x0078)

**Table 20-27. Raw Blending Coefficients Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:6] | SI_ABD_DCOE | R/W | 0x30 | D coefficient of the Alpha blending operation between the current input pixel and the pixel stored in the memory.<br>Px * pCoe (1.4) + Dk * dCoe (s1.4; 2's complement) |
| [15:5] | - | - | - | Reserved |
| [4:0] | SI_ABD_PCOE | R/W | 0x10 | P coefficient of the Alpha blending operation between the current input pixel and the pixel stored in the memory.<br>Px * pCoe (1.4) + Dk * dCoe |

### 20.4.2.26 Dark Raw Precision Register (Offset: 0x007C)

**Table 20-28. Dark Raw Precision Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:16] | DFS_HI_TH | R/W | 0xFF | The pixel values, which is below the threshold, stored in the memory will be treated as the dark pixel; otherwise, it will be a bad pixel.<br>Format: 8. |
| [15:2] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [1:0] | DFS_PBIT | R/W | 1 | Precision dark raw data<br>0: Raw data [9:2]<br>1: Raw data [10:3]<br>2: Raw data [11:4]<br>3: Reserved |

### 20.4.2.27 Defect Pixel Correction Control Register (Offset: 0x0080)

**Table 20-29. Defect Pixel Correction Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:5] | - | - | - | Reserved |
| 4 | DPC_RCV_MODE | R/W | 0 | Defect pixel correction in the pixel-recovery mode<br>0: Pixel recovery checked using neighbor two pixels<br>1: Pixel recovery checked using center and neighbor two pixels |
| [3:2] | - | - | - | Reserved |
| [1:0] | DPC_MODE | R/W | 1 | Defect pixel correction mode<br>0, 2: Reserved<br>1: Auto mode only<br>3: Auto and pre-checked by DFS |

### 20.4.2.28 Defect Pixel Correction Threshold Register (Offset: 0x0084)

**Table 20-30. Defect Pixel Correction Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | DPC_TH3 | R/W | 0x4 | Count threshold of the defect pixel correction |
| [27:24] | - | - | - | Reserved |
| [23:16] | DPC_TH2 | R/W | 0 | Threshold 2 of the defect pixel correction |
| [15:8] | - | - | - | Reserved |
| [7:0] | DPC_TH1 | R/W | 0 | Threshold 1 of the defect pixel correction |

### 20.4.2.29 Crosstalk Threshold Register (Offset: 0x0088)

**Table 20-31. Crosstalk Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | CTK_TH4 | R/W | 0 | Threshold 4 of Gb, Gr crosstalk delta filter |
| [23:16] | CTK_TH3 | R/W | 0 | Threshold 3 of Gb, Gr crosstalk delta filter |
| [15:8] | CTK_TH2 | R/W | 0 | Threshold 2 of Gb, Gr crosstalk delta filter |
| [7:0] | CTK_TH1 | R/W | 0 | Threshold 1 of Gb, Gr crosstalk delta filter |

### 20.4.2.30 Crosstalk Weighting Register (Offset: 0x008C)

**Table 20-32. Crosstalk Weighting Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:24] | CTK_GMODE | R/W | 3 | Crosstalk removal setting on different types of G cells<br>0, 3: Crosstalk removal operating on Gb and Gr<br>1: Crosstalk removal operating on Gb only<br>2: Crosstalk removal operating on Gr only |
| [23:19] | - | - | - | Reserved |
| [18:16] | CTK_THRAN | R/W | 0 | Crosstalk delta threshold range |
| [15:12] | - | - | - | Reserved |
| [11:8] | CTK_W3 | R/W | 0 | Crosstalk delta shift weighting 3 |
| [7:4] | CTK_W2 | R/W | 0 | Crosstalk delta shift weighting 2 |
| [3:0] | CTK_W1 | R/W | 0 | Crosstalk delta shift weighting 1 |

### 20.4.2.31 RB Pre-gain Register (Offset: 0x0090)

**Table 20-33. RB Pre-gain Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_B_GAIN | R/W | 0x400 | Pre-gain for the B cell on the GB/BG channel<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [11:0] | RAW_R_GAIN | R/W | 0x400 | Pre-gain for the R cell on the GR/RG channel<br>Format: 2.10 |

### 20.4.2.32 G Pre-gain Register (Offset: 0x0094)

**Table 20-34. G Pre-gain Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_BG_GAIN | R/W | 0x400 | Pre-gain for the G cell on the GB/BG channel<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | RAW_RG_GAIN | R/W | 0x400 | Pre-gain for the G cell on the GR/RG channel<br>Format: 2.10 |

### 20.4.2.33 RB Gain Register (Offset: 0x0098)

**Table 20-35. RB Gain Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | B_GAIN | R/W | 0x400 | B gain on RGB color space<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | R_GAIN | R/W | 0x400 | R gain on RGB color space<br>Format: 2.10 |

### 20.4.2.34 G Gain Register (Offset: 0x009C)

**Table 20-36. G Gain Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | G_GAIN | R/W | 0x400 | G gain on the RGB color space<br>Format: 2.10 |

### 20.4.2.35 Color Correction Matrix 00 Register (Offset: 0x00A0)

**Table 20-37. Color Correction Matrix 00 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CC_MX01 | R/W | 0 | Color correction matrix coefficient 01<br>Format: s1.10 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CC_MX00 | R/W | 0x400 | Color correction matrix coefficient 00<br>Format: s1.10 (2's complement) |

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \left( \begin{bmatrix} cc00 & cc01 & cc02 \\ cc10 & cc11 & cc12 \\ cc20 & cc21 & cc22 \end{bmatrix} \right) \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} cc\_offset0 \\ cc\_offset1 \\ cc\_offset2 \end{bmatrix}$$

**Figure 20-2. Color Correction Matrix**

### 20.4.2.36 Color Correction Matrix 01 Register (Offset: 0x00A4)

**Table 20-38. Color Correction Matrix 01 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | CC_OFFSET0 | R/W | 0 | Color correction matrix offset 0<br>Format: s6 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CC_MX02 | R/W | 0 | Color correction matrix coefficient 02<br>Format: s1.10 (2's complement) |

### 20.4.2.37 Color Correction Matrix 10 Register (Offset: 0x00A8)

**Table 20-39. Color Correction Matrix 10 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CC_MX11 | R/W | 0x400 | Color correction matrix coefficient 11<br>Format: s1.10 (2's complement) |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [15:12] | - | - | - | Reserved |
| [11:0] | CC_MX10 | R/W | 0x400 | Color correction matrix coefficient 10<br>Format: s1.10 (2's complement) |

### 20.4.2.38 Color Correction Matrix 11 Register (Offset: 0x00AC)

**Table 20-40. Color Correction Matrix 11 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | CC_OFFSET1 | R/W | 0 | Color correction matrix offset 1<br>Format: s6 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CC_MX12 | R/W | 0 | Color correction matrix coefficient 12<br>Format: s1.10 (2's complement) |

### 20.4.2.39 Color Correction Matrix 20 Register (Offset: 0x00B0)

**Table 20-41. Color Correction Matrix 20 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CC_MX21 | R/W | 0 | Color correction matrix coefficient 21<br>Format: s1.10 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CC_MX20 | R/W | 0x400 | Color correction matrix coefficient 20<br>Format: s1.10 (2's complement) |

### 20.4.2.40 Color Correction Matrix 21 Register (Offset: 0x00B4)

**Table 20-42. Color Correction Matrix 21 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [22:16] | CC_OFFSET2 | R/W | 0 | Color correction matrix offset 2<br>Format: s6 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CC_MX22 | R/W | 0 | Color correction matrix coefficient 22.<br>Format: s1.10 (2's complement) |

### 20.4.2.41 Color Space Conversion Matrix 00 Register (Offset: 0x00B8)

**Table 20-43. Color Space Conversion Matrix 00 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CV_MX01 | R/W | 0xE54 | Color space conversion matrix coefficient 01<br>Format: s1.10 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CV_MX00 | R/W | 0x200 | Color space conversion matrix coefficient 00<br>Format: s1.10 (2's complement) |

### 20.4.2.42 Color Space Conversion Matrix 01 Register (Offset: 0x00BC)

**Table 20-44. Color Space Conversion Matrix 01 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | CV_MX02 | R/W | 0xFAD | Color space conversion matrix coefficient 02<br>Format: s1.10 (2's complement) |

### 20.4.2.43 Color Space Conversion Matrix 10 Register (Offset: 0x00C0)

**Table 20-45. Color Space Conversion Matrix 10 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CV_MX11 | R/W | 0x259 | Color space conversion matrix coefficient 11<br>Format: s1.10 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CV_MX10 | R/W | 0x132 | Color space conversion matrix coefficient 10<br>Format: s1.10 (2's complement) |

### 20.4.2.44 Color Space Conversion Matrix 11 Register (Offset: 0x00C4)

**Table 20-46. Color Space Conversion Matrix 11 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | CV_MX12 | R/W | 0x74 | Color space conversion matrix coefficient 12<br>Format: s1.10 (2's complement) |

### 20.4.2.45 Color Space Conversion Matrix 20 Register (Offset: 0x00C8)

**Table 20-47. Color Space Conversion Matrix 20 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | CV_MX21 | R/W | 0xEAD | Color space conversion matrix coefficient 21<br>Format: s1.10 (2's complement) |
| [15:12] | - | - | - | Reserved |
| [11:0] | CV_MX20 | R/W | 0xF54 | Color space conversion matrix coefficient 20<br>Format: s1.10 (2's complement) |

### 20.4.2.46 Color Space Conversion Matrix 01 Register (Offset: 0x00CC)

**Table 20-48. Color Space Conversion Matrix 21 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | CV_MX22 | R/W | 0x200 | Color space conversion matrix coefficient 22<br>Format: s1.10 (2's complement) |

### 20.4.2.47 Contrast Enhancement Control 0 Register (Offset: 0x00D0)

**Table 20-49. Contrast Enhancement Control 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | CE_CGOFST | R/W | 0 | CE CbCr gain offset<br>Format: s1.6 (2's complement) |
| [23:16] | CE_YGTH | R/W | 0x20 | CE Ygain threshold to determine the damping factor of CbCr<br>Format: 3.5 |
| [15:2] | - | - | - | Reserved |
| [1:0] | CI_THRATIO | R/W | 0x2 | CI threshold ratio<br>0: Ratio = 0.25<br>1: Ratio = 0.5<br>2: Ratio = 0.75<br>3: Ratio = 1 |

### 20.4.2.48 Contrast Enhancement Control 1 Register (Offset: 0x00D4)

**Table 20-50. Contrast Enhancement Control 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | CE_CDMP2 | R/W | 0xFF | CE CbCr gain damping factor 2<br>Format: .8 |
| [23:16] | CE_CDMP1 | R/W | 0xFF | CE CbCr gain damping factor 1<br>Format: .8 |
| [15:13] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [12:8] | CE_RANGE_FACT | R/W | 0 | Contrast enhancement Luma range factor to control range function<br>0 ~ 0x1f: Higher value causes lower range of the function effect |
| [7:3] | - | - | - | Reserved |
| [2:0] | CE_LEV | R/W | - | Contrast enhancement of the strength control<br>0 ~ 7: Higher value causes lower strength |

### 20.4.2.49 Saturation Control Register (Offset: 0x00D8)

**Table 20-51. Saturation Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| 24 | YCC_CCUVX2 | R/W | 0 | Cb/Cr curve reference selection<br>0: Cb/Cr refers to same curve with 64 segments.<br>1: Cb/Cr refers to different curves with 32 segments. |
| [23:21] | - | - | - | Reserved |
| 20 | IE_SAT_POL | R/W | 0 | Color saturation increasing/decreasing control<br>0: Increase the color saturation<br>1: Decrease the color saturation |
| [19:17] | - | - | - | Reserved |
| 16 | IE_WEIGHT_EN | R/W | 1 | Image enhancement Hue related weighting function switch<br>0: Disable Hue weighting factor<br>1: Enable Hue weighting factor |
| [15:5] | - | - | - | Reserved |
| 4 | CS_Y_SRC_SEL | R/W | 0 | CS luma source selection<br>0: CS Y source is before gamma correction.<br>1: CS Y source is after gamma correction. |
| [3:1] | - | - | - | Reserved |
| 0 | CS_WEIGHT_EN | R/W | 1 | Chroma suppression of the saturation related weighting function switch<br>0: Disable the saturation weighting factor<br>1: Enable the saturation weighting factor |

### 20.4.2.50 Sharpness Control Register (Offset: 0x00DC)

**Table 20-52. Sharpness Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | SP_EDGE_TH | R/W | 0 | SP edge detection threshold |
| [23:20] | - | - | - | Reserved |
| 19 | SP_DMP_EN | R/W | 0 | SP edge overshoot and undershoot damping function switch<br>0: Enable the SP edge damping<br>1: Disable the SP edge damping |
| 18 | - | - | - | Reserved |
| [17:16] | SP_DMP_LVL | R/W | 0 | SP edge overshoot and undershoot damping level<br>0: 25% damping<br>1: 50% damping<br>2: 75% damping<br>3: 100% damping |
| [15:7] | - | - | - | Reserved |
| [6:0] | YCC_Y_OFFSET | R/W | 0 | Luma offset after Y compensation curve<br>Format: s6 (2's complement) |

### 20.4.2.51 FCS Control Register (Offset: 0x00E0)

**Table 20-53. FCS Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:24] | FCS_SGTH_DELTA | R/W | 0x3 | Delta value for the high/low threshold of the FCS strong irregular color<br>0: Delta value = 0<br>1: Delta value = 3/4 * average deviation<br>2: Delta value = 1 * average deviation<br>3: Delta value = 3/2 * average deviation<br>4: Delta value = 2 * average deviation<br>5: Delta value = 3 * average deviation<br>6: Delta value = 4 * average deviation |
| [23:22] | - | - | - | Reserved |
| [21:8] | FCS_WK_TH | R/W | 0x19 | FCS weak irregular color threshold |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [7:1] | - | - | - | Reserved |
| 0 | FCS_EDGE_TH | R/W | 1 | Edge condition switch for the FCS operation<br>0: Don't care the edge level of the FCS operation<br>1: Take edge-qualified pixel into the FCS operation |

### 20.4.2.52 HBLANK Extended Register (Offset: 0x00EC)

**Table 20-54. HBLANK Extended Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | EXT1_HBLK_PIX | R/W | 0x100 | Horizontal blanking width for the extended line of image pipeline 1 |
| [15:10] | - | - | - | Reserved |
| [9:0] | EXT0_HBLK_PIX | R/W | 0x100 | Horizontal blanking width for the extended line of image pipeline 0 |

### 20.4.2.53 AE & AWB Control Register (Offset: 0x0100)

**Table 20-55. AE & AWB Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| 26 | AWB_STA_MODE1 | R/W | 0 | Behavior of the AWB accumulator when the source exceeds the AWB high/low & color saturation threshold<br>0: No accumulation on unqualified pixels<br>1: Accumulates all sampling pixels |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [25:24] | AWB_STA_MODE0 | R/W | 0 | Types of the AWB statistics for extracting the raw data<br>0: One pixel<br>1: Average 4 pixels<br>2: R, G, B Summation and number of qualified pixels (1/2 sub-sampling)<br>Max pixel numbers per window = 4095<br>Format: 20. 20. 20. 12.<br>3: (R/G, B/G, R/B) chromaticity, G average, number of qualified pixels<br>Max pixel numbers per window = 4095<br>Format: (8.8, 8.8, 8.8), 8.4, 12. |
| [23:22] | - | - | - | Reserved |
| 21 | AWB_OVERWR | R/W | 0 | Control the action of storing AWB STA to wait the RDY bit to be cleared or not.<br>0: Wait clearing the RDY bit<br>1: Overwrite the AWB STA unconditionally |
| 20 | AWB_INT_STYLE | R/W | 1 | Interrupt style of AWB statistics ready<br>0: Interrupt per quarter of AWB window<br>1: Interrupt by full AWB window statistics |
| [19:16] | - | - | - | Reserved |
| [15:14] | AE_STA_QFDEF | R/W | 0 | High, low qualified threshold definition<br>0: ae_luma_nlo_th: low threshold<br>ae_luma_nhi_th: high threshold<br>1: ae_luma_vlo_th: low threshold<br>ae_luma_nhi_th: high threshold<br>2: ae_luma_nlo_th: low threshold<br>ae_luma_vhi_th: high threshold<br>3: ae_luma_vlo_th: low threshold<br>ae_luma_vhi_th: high threshold |
| [13:12] | - | - | - | Reserved |
| [11:10] | AE_STA_MODE1 | R/W | 0 | Behavior of the AE accumulator when the source Luminance is exceeded than AE high/low threshold<br>0: No accumulation on unqualified pixels<br>1: Accumulates all sampling pixels<br>2: Clip unqualified pixel value to threshold value before Y accumulation |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| | | | | 3: Reserved |
| 9 | - | - | - | Reserved |
| 8 | AE_STA_MODE0 | R/W | 0 | Type of AE statistics data<br>0: R, G, and B Summation<br>1: Y summation and high, low threshold pixel counts |
| [7:6] | - | - | - | Reserved |
| 5 | AE_OVERWR | R/W | 0 | Control the action of storing AE STA to wait RDY bit clearing or not<br>0: Waiting the clearing of RDY bit<br>1: Overwrite the AE STA unconditionally |
| 4 | AE_INT_STYLE | R/W | 1 | Interrupt style of AE statistics ready<br>0: Interrupt per quarter of AE window<br>1: Interrupt by full AE window statistics |
| [3:2] | - | - | - | Reserved |
| [1:0] | AE_SRC_SEL | R/W | 0 | Source image selection for the AE statistics<br>0: AE statistic data before the gamma correction<br>1: AE statistic data after the gamma correction<br>2: AE statistic data after YCC[7:0] |

### 20.4.2.54 AF & MD Control Register (Offset: 0x0104)

**Table 20-56. AF & MD Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:21] | - | - | - | Reserved |
| 20 | MD_INT_STYLE | R/W | 1 | Interrupt style of MD statistics ready<br>0: Interrupt per quarter of MD window<br>1: Interrupt by full MD window statistics |
| [19:18] | - | - | - | Reserved |
| [17:16] | MD_SRC_SEL | R/W | 0 | Source data selection for MD statistics<br>0: Y<br>1: Cb<br>2: Cr<br>3: Reserved |
| [15:6] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 5 | AF_OVERWR | R/W | 0 | Control the action of storing AF STA to wait the RDY bit to be cleared or not<br>0: Wait clearing of the RDY bit<br>1: Overwrite the AF STA unconditionally |
| 4 | AF_INT_STYLE | R/W | 0 | Interrupt style of AF statistics ready<br>0: Interrupt per AF window row<br>1: Interrupt by full AF window statistics |
| [3:2] | - | - | - | Reserved |
| [1:0] | AF_SRC_SEL | R/W | 1 | 0, 3: Reserved<br>1: Y before CI<br>2: G before CI |

### 20.4.2.55 HSTG Source Register (Offset: 0x0108)

**Table 20-57. HSTG Source Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:4] | - | - | - | Reserved |
| [3:0] | | R/W | 0 | Data selection for Histogram statistics<br>0: Y after CV<br>1: Y after CE<br>2: Y before CI<br>3: Reserved<br>4: R before CC<br>5: G before CC<br>6: B before CC<br>7: Reserved<br>8: Bayer Raw R<br>9: Bayer Raw B<br>0xA: Bayer Raw Gr<br>0xB: Bayer Raw Gb<br>0xC ~ 0xF: Reserved |

### 20.4.2.56 AE Window Start Point Register (Offset: 0x0110)

**Table 20-58. AE Window Start Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AE_WIN_Y | R/W | 0 | Start line of the AE window |
| [15:12] | - | - | - | Reserved |
| [11:0] | AE_WIN_X | R/W | 0 | Start pixel of the AE window |

### 20.4.2.57 AE Window Size Register (Offset: 0x0114)

**Table 20-59. AE Window Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | AE_WIN_Y_SIZE | R/W | 0 | Height of the AE window<br>Max height: 256 |
| [15:9] | - | - | - | Reserved |
| [8:0] | AE_WIN_X_SIZE | R/W | 0 | Width of the AE window<br>Max width: 256 |

### 20.4.2.58 AE Window Number Register (Offset: 0x0118)

**Table 20-60. AE Window Number Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | AE_WIN_Y_NUM | R/W | 0 | Number of the vertical AE windows<br>Max: 32 |
| [15:6] | - | - | - | Reserved |
| [5:0] | AE_WIN_X_NUM | R/W | 0 | Number of the horizontal AE windows<br>Max: 32 |

### 20.4.2.59 AE Window Gap Register (Offset: 0x011C)

**Table 20-61. AE Window Gap Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | AE_WIN_Y_GAP | R/W | 0 | Gap size between the AE vertical windows |
| [15:0] | - | - | - | Reserved |

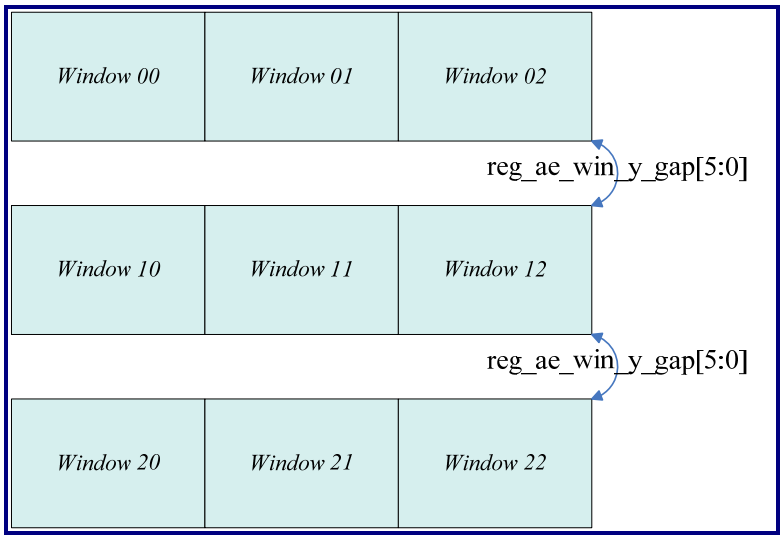



**Figure 20-3. AE STA Window Gap**

### 20.4.2.60 AE STA Threshold Register (Offset: 0x0120)

**Table 20-62. AE STA Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | AE_LUMA_VHI_TH | R/W | 0xFF | AE luminance threshold to define very bright pixel |
| [23:16] | AE_LUMA_NHI_TH | R/W | 0xFF | AE luminance threshold to define normal white pixel |
| [15:8] | AE_LUMA_VLO_TH | R/W | 0 | AE luminance threshold to define very dark pixel |
| [7:0] | AE_LUMA_NLO_TH | R/W | 0 | AE luminance threshold to define normal black pixel |

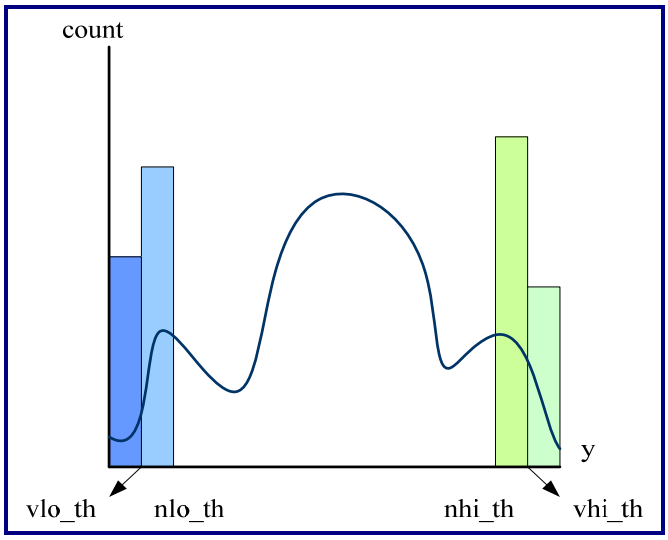



**Figure 20-4. AE STA Threshold**

## 20.4.2.61 AWB Window Start Point Register (Offset: 0x0140)

**Table 20-63. AWB Window Start Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AWB_WIN_Y | R/W | 0 | Start line of the AWB window<br>Only the even number is permitted. |
| [15:12] | - | - | - | Reserved |
| [11:0] | AWB_WIN_X | R/W | 0 | Start pixel of the AWB window<br>Only the even number is permitted. |

## 20.4.2.62 AWB Window Size Register (Offset: 0x0144)

**Table 20-64. AWB Window Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | AWB_WIN_Y_SIZE | R/W | 0 | Height of the AWB window<br>Max height: 256<br>Only the even number is permitted. |
| [15:9] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [8:0] | AWB_WIN_X_SIZE | R/W | 0 | Width of the AWB window<br>Max width: 256<br>Only the even number is permitted. |

### 20.4.2.63 AWB Window Number Register (Offset: 0x0148)

**Table 20-65. AWB Window Number Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | AWB_WIN_Y_NUM | R/W | 0 | Number of the vertical AWB windows<br>Max: 64 |
| [15:8] | - | - | - | Reserved |
| [7:0] | AWB_WIN_X_NUM | R/W | 0 | Number of the horizontal AWB windows<br>Max: 128 |

### 20.4.2.64 AWB Window Gap Register (Offset: 0x014C)

**Table 20-66. AWB Window Gap Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | AWB_WIN_Y_GAP | R/W | 0 | Gap size between the AWB vertical windows<br>Only even number is permitted. |
| [15:0] | - | - | - | Reserved |

### 20.4.2.65 AWB STA Threshold Register (Offset: 0x0150)

**Table 20-67. AWB STA Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| [15:8] | AWB_LUMA_HI_TH | R/W | 0xFF | Luma high threshold to qualify pixels for AWB statistics |
| [7:0] | AWB_LUMA_LO_TH | R/W | 0 | Luma low threshold to qualify pixels for AWB statistics |

### 20.4.2.66 AF Pre-Curve Point Register (Offset: 0x0170)

**Table 20-68. AF Pre-Curve Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | AF_Y2_PTR | R/W | 0x80 | AF pre-curve output 2 |
| [23:16] | AF_Y1_PTR | R/W | 0x40 | AF pre-curve output 1 |
| [15:8] | AF_X2_PTR | R/W | 0x80 | AF pre-curve input 2 |
| [7:0] | AF_X1_PTR | R/W | 0x40 | AF pre-curve input 1 |

### 20.4.2.67 AF Pre-Curve Slope 0 Register (Offset: 0x0174)

**Table 20-69. AF Pre-Curve Slope 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AF_S2_SLOPE | R/W | 0x100 | AF pre-curve segment 2 slope<br>Format: 4.8 |
| [15:12] | - | - | - | Reserved |
| [11:0] | AF_S1_SLOPE | R/W | 0x100 | AF pre-curve segment 1 slope<br>Format: 4.8 |

### 20.4.2.68 AF Pre-Curve Slope 1 Register (Offset: 0x0178)

**Table 20-70. AF Pre-Curve Slope 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AF_S3_SLOPE | R/W | 0x100 | AF pre-curve segment 3 slope<br>Format: 4.8 |

### 20.4.2.69 AF Window Start Point Register (Offset: 0x0180)

**Table 20-71. AF Window Start Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AF_WIN_Y | R/W | 0 | Start line of the AF window |
| [15:12] | - | - | - | Reserved |
| [11:0] | AF_WIN_X | R/W | 0 | Start pixel of the AF window |

### 20.4.2.70 AF Window Size Register (Offset: 0x0184)

**Table 20-72. AF Window Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | AF_WIN_Y_SIZE | R/W | 0 | Height of the AF window |
| [15:9] | - | - | - | Reserved |
| [8:0] | AF_WIN_X_SIZE | R/W | 0 | Width of the AF window<br>Min: 16 |

### 20.4.2.71 AF Window Number Register (Offset: 0x0188)

**Table 20-73. AF Window Number Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:19] | - | - | - | Reserved |
| [18:16] | AF_WIN_Y_NUM | R/W | 0 | Number of the AFvertical windows<br>Max: 5 |
| [15:3] | - | - | - | Reserved |
| [2:0] | AF_WIN_X_NUM | R/W | 0 | Number of the AF horizontal windows<br>Max: 5 |

### 20.4.2.72 AF Window Gap Register (Offset: 0x018C)

**Table 20-74. AF Window Gap Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | AF_WIN_Y_GAP | R/W | 0 | Gap between the AF vertical windows |
| [15:9] | - | - | - | Reserved |
| [8:0] | AF_WIN_X_GAP | R/W | 0x4 | Gap between the AF horizontal windows<br>Min: 4 |

### 20.4.2.73 AF HF1 Threshold Register (Offset: 0x0190)

**Table 20-75. AF HF1 Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | AF_HACC1_SFT | R/W | 0 | AF horizontal accumulator 1 output shifter<br>0: Keep the original horizontal accumulator 1 result<br>0x1 ~ 0xF: The result of the horizontal accumulator 1 is shifted to the right by 1 ~ 15 bits. |
| [27:24] | AF_HF1_SFT | R/W | 0 | AF horizontal filter 1 output shifter<br>0: Keep the original horizontal filter 1 result<br>0x1 ~ 0xF: The result of the horizontal filter 1 is shifted to the right by 1 ~ 15 bits. |
| [23:20] | - | - | - | Reserved |
| [19:0] | AF_HF1_TH | R/W | 0 | AF horizontal filter 1 threshold |

### 20.4.2.74 AF HF2 Threshold Register (Offset: 0x0194)

**Table 20-76. AF HF2 Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | AF_HACC2_SFT | R/W | 0 | AF horizontal accumulator 2 output shifter<br>0: Keep the original horizontal Accumulator 2 result<br>0x1 ~ 0xF: The result of the horizontal accumulator 2 is shifted to the right by 1 ~ 15 bits. |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [27:24] | AF_HF2_SFT | R/W | 0 | AF horizontal filter 2 output shifter<br>0: Keep the original horizontal filter 2 result<br>0x1 ~ 0xF: The result of the horizontal filter 2 is shifted to the right by 1 ~ 15 bits. |
| [23:20] | - | - | - | Reserved |
| [19:0] | AF_HF2_TH | R/W | 0 | AF horizontal filter 2 threshold |

### 20.4.2.75 AF VF Threshold Register (Offset: 0x0198)

**Table 20-77. AF VF Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | AF_VACC_SFT | R/W | 0 | AF vertical Accumulator output Shifter<br>0: Keep the original vertical accumulator result<br>0x1 ~ 0xF: The result of the vertical accumulator is shifted to the right by 1 ~ 15 bits |
| [27:24] | AF_VF_SFT | R/W | 0 | AF vertical filter output shifter<br>0: Keep the original vertical filter result<br>0x1 ~ 0xF: The result of the vertical filter is shifted to the right by 1 ~ 15 bits |
| [23:21] | - | - | - | Reserved |
| [20:0] | AF_VF_TH | R/W | 0 | AF vertical filter threshold |

### 20.4.2.76 AF HF1 Coefficient 01 Register (Offset: 0x01A0)

**Table 20-78. AF HF1 Coefficient 01Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:16] | AF_HF1_COE1 | R/W | 0 | AF horizontal filter 1 coefficient 1<br>Format: s7 (2's complement) |
| [15:7] | - | - | - | Reserved |
| [6:0] | AF_HF1_COE0 | R/W | 0 | AF horizontal filter 1 coefficient 0<br>Format: s6 (2's complement) |

### 20.4.2.77 AF HF1 Coefficient 23 Register (Offset: 0x01A4)

**Table 20-79. AF HF1 Coefficient 23 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | AF_HF1_COE3 | R/W | 0 | AF horizontal filter 1 coefficient 3<br>Format: s9 (2's complement) |
| [15:9] | - | - | - | Reserved |
| [8:0] | AF_HF1_COE2 | R/W | 0 | AF horizontal filter 1 coefficient 2<br>Format: s8 (2's complement) |

### 20.4.2.78 AF HF1 Coefficient 45 Register (Offset: 0x01A8)

**Table 20-80. AF HF1 Coefficient 45 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AF_HF1_COE5 | R/W | 0 | AF horizontal filter 1 coefficient 5<br>Format: s11 (2's complement) |
| [15:11] | - | - | - | Reserved |
| [10:0] | AF_HF1_COE4 | R/W | 0 | AF horizontal filter 1 coefficient 4<br>Format: s10 (2's complement) |

### 20.4.2.79 AF HF1 Coefficient 67 Register (Offset: 0x01AC)

**Table 20-81. AF HF1 Coefficient 67 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:16] | AF_HF1_COE7 | R/W | 0 | AF horizontal filter 1 coefficient 7<br>Format: s13 (2's complement) |
| [15:13] | - | - | - | Reserved |
| [12:0] | AF_HF1_COE6 | R/W | 0 | AF horizontal filter 1 coefficient 6<br>Format: s12 (2's complement) |

### 20.4.2.80 AF HF2 Coefficient 01 Register (Offset: 0x01B0)

**Table 20-82. AF HF2 Coefficient 01Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:16] | AF_HF2_COE1 | R/W | 0 | AF horizontal filter 2 coefficient 1<br>Format: s7 (2's complement) |
| [15:7] | - | - | - | Reserved |
| [6:0] | AF_HF2_COE0 | R/W | 0 | AF horizontal filter 2 coefficient 0<br>Format: s6 (2's complement) |

### 20.4.2.81 AF HF2 Coefficient 23 Register (Offset: 0x01B4)

**Table 20-83. AF HF2 Coefficient 23 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:16] | AF_HF2_COE3 | R/W | 0 | AF horizontal filter 2 coefficient 3<br>Format: s9 (2's complement) |
| [15:9] | - | - | - | Reserved |
| [8:0] | AF_HF2_COE2 | R/W | 0 | AF horizontal filter 2 coefficient 2<br>Format: s8 (2's complement) |

### 20.4.2.82 AF HF2 Coefficient 45 Register (Offset: 0x01B8)

**Table 20-84. AF HF2 Coefficient 45 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AF_HF2_COE5 | R/W | 0 | AF horizontal filter 2 coefficient 5<br>Format: s11 (2's complement) |
| [15:11] | - | - | - | Reserved |
| [10:0] | AF_HF2_COE4 | R/W | 0 | AF horizontal filter 2 coefficient 4<br>Format: s10 (2's complement) |

### 20.4.2.83 AF HF2 Coefficient 67 Register (Offset: 0x01BC)

**Table 20-85. AF HF2 Coefficient 67 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:16] | AF_HF2_COE7 | R/W | 0 | AF horizontal filter 2 coefficient 7<br>Format: s13 (2's complement) |
| [15:13] | - | - | - | Reserved |
| [12:0] | AF_HF2_COE6 | R/W | 0 | AF horizontal filter 2 coefficient 6<br>Format: s12 (2's complement) |

### 20.4.2.84 AF VF Coefficient 01 Register (Offset: 0x01C0)

**Table 20-86. AF VF Coefficient 01 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | AF_VF_COE1 | R/W | 0 | AF vertical filter coefficient 1<br>Format: s8 (2's complement) |
| [15:7] | - | - | - | Reserved |
| [6:0] | AF_VF_COE0 | R/W | 0 | AF vertical filter coefficient 0<br>Format: s6 (2's complement) |

### 20.4.2.85 AF VF Coefficient 23 Register (Offset: 0x01C4)

**Table 20-87. AF VF Coefficient 23 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | AF_VF_COE3 | R/W | 0 | AF vertical filter coefficient 3<br>Format: s11 (2's complement) |
| [15:11] | - | - | - | Reserved |
| [10:0] | AF_VF_COE2 | R/W | 0 | AF vertical filter coefficient 2<br>Format: s10 (2's complement) |

### 20.4.2.86 AF VF Coefficient 4 Register (Offset: 0x01C8)

**Table 20-88. AF VF Coefficient 4 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:15] | - | - | - | Reserved |
| [14:0] | AF_VF_COE4 | R/W | 0 | AF vertical filter coefficient 4<br>Format: 15 |

### 20.4.2.87 MD Window Start Point Register (Offset: 0x01D0)

**Table 20-89. MD Window Start Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31: 28] | - | - | - | Reserved |
| [27:16] | MD_WIN_Y | R/W | 0 | Start line of the MD window |
| [15:12] | - | - | - | Reserved |
| [11:0] | MD_WIN_X | R/W | 0 | Start pixel of the MD window |

### 20.4.2.88 MD Window Size Register (Offset: 0x01D4)

**Table 20-90. MD Window Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | MD_WIN_Y_SIZE | R/W | 0 | Height of the MD window<br>Max: 32 |
| [15:6] | - | - | - | Reserved |
| [5:0] | MD_WIN_X_SIZE | R/W | 0 | Width of the MD window<br>Min: 32 |

### 20.4.2.89 MD Window Number Register (Offset: 0x01D8)

**Table 20-91. MD Window Number Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:16] | MD_WIN_Y_NUM | R/W | 0 | Number of the vertical MD windows<br>Max.: 128 |
| [15:8] | - | - | - | Reserved |
| [7:0] | MD_WIN_X_NUM | R/W | 0 | Number of the horizontal MD windows<br>Max.: 128 |

### 20.4.2.90 MD Window Gap Register (Offset: 0x01DC)

**Table 20-92. MD Window Gap Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | MD_WIN_Y_GAP | R/W | 0 | Gap between the MD vertical windows |
| [15:0] | - | - | - | Reserved |

### 20.4.2.91 MD Control 0 Register (Offset: 0x01E0)

**Table 20-93. MD Control 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| 24 | MD_SHAD_EN | R/W | 1 | Do shadow detection |
| [23:16] | MD_SHAD_TH | R/W | 0xCC | Shadow threshold<br>The shadow will be detected if the pixel is darker than the background. |
| [15:8] | - | - | - | Reserved |
| [7:0] | MD_INIT_VAL | R/W | 7 | Initial value for the mixture of the Gaussian operation |

### 20.4.2.92 MD Control 1 Register (Offset: 0x01E4)

**Table 20-94. MD Control 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | - | - | - | Reserved |
| [30:16] | MD_B_ALPHA | R/W | 0x7FE0 | 2^15 - alpha |
| 15 | - | - | - | Reserved |
| [14:0] | MD_A_ALPHA | R/W | 0x20 | Update speed of the MD statistic<br>alpha = 1/T<br>T: Time interval of the data average |

### 20.4.2.93 MD Control 2 Register (Offset: 0x01E8)

**Table 20-95. MD Control 2 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:8] | MD_PRUNE | R/W | 0xF | Prune = -(alpha * CT)/256<br>CT: Complexity of reduction prior<br>It is related to the satisfied condition of one MD component found. |
| [7:5] | - | - | - | Reserved |
| [4:0] | MD_SIGMA | R/W | 0xB | Initial value of the standard deviation for the newly generated component.<br>It affects the algorithm adaptation speed. |

### 20.4.2.94 MD Control 3 Register (Offset: 0x01EC)

**Table 20-96. MD Control 3 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | MD_TBG_TH | R/W | 0x1CCB | Threshold for the component merged into the background model |
| [15:12] | - | - | - | Reserved |
| [11:8] | MD_SMG_TH | R/W | 0x9 | Decode the squared Mahalan distance threshold if a sample is close to the existing components |
| [7:4] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [3:0] | MD_SMB_TH | R/W | 0x9 | Decide the squared Mahalan distance threshold if it is well described by the background model. |

### 20.4.2.95 HSTG Window Start Point Register (Offset: 0x01F0)

**Table 20-97. HSTG Window Start Point Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | HSTG_WIN_Y | R/W | 0 | Start line of the Histogram window |
| [15:12] | - | - | - | Reserved |
| [11:0] | HSTG_WIN_X | R/W | 0 | Start pixel of the Histogram window |

### 20.4.2.96 HSTG Window Size Register (Offset: 0x01F4)

**Table 20-98. HSTG Window Size Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | HSTG_WIN_Y_SIZE | R/W | 0 | Height of the Histogram window |
| [15:12] | - | - | - | Reserved |
| [11:0] | HSTG_WIN_X_SIZE | R/W | 0 | Width of the Histogram window |

### 20.4.2.97 DMA Control Register (Offset: 0x0200)

**Table 20-99. DMA Control Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [26:24] | AHB_HRESP_DLY | R/W | 0 | AHB retry/split delay clock = 8+ reg_hresp_dly |
| 23 | DMA_MST_EN | R/W | 0 | Single or ping-pong frame buffer mode switch for Image DMA<br>0: Upstream DMA always writes to the frame buffer 0.<br>1: Enter the ping-pong frame buffer mode |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 22 | STADMA_MST_EN | R/W | 0 | Single or ping-pong frame buffer mode switch for STA DMA<br>0: STA DMA always writes to the frame buffer 0.<br>1: Enter the ping-pong frame buffer mode |
| [21:18] | - | - | - | Reserved |
| 17 | DMA_V_FLIP | R/W | 0 | Up-stream DMA vertical flipping mode switch<br>It is necessary to program the destination base address for each flipping mode.<br>0: No flipping on the DMA vertical direction<br>1: Do flipping on the DMA vertical direction |
| 16 | DMA_H_FLIP | R/W | 0 | Up-stream DMA horizontal flipping mode switch<br>It is necessary to program the destination base address for each flipping mode<br>0: No mirror on DMA horizontal direction<br>1: Do mirror on DMA horizontal direction |
| [15:10] | - | - | - | Reserved |
| [9:8] | DMA_SPLIT | R/W | 0 | YCbCr data arrangement in memory<br>0: Packet format<br>YCbCr data are written into one frame buffer.<br>1: Planar format<br>Write three color components into three different memory buffers |
| [7:6] | DMA_C_WLEN | R/W | 0x3 | Cb,Cr burst write length<br>0: 1 data package for one burst<br>1: 2 data packages for one burst<br>2: 4 data packages for one burst<br>3: 8 data packages for one burst |
| [5:4] | DMA_Y_WLEN | R/W | 0x3 | Y burst write length<br>0: 4 data packages for one burst<br>1: 8 data packages for one burst<br>2: 12 data packages for one burst<br>3: 16 data packages for one burst |
| [3:2] | DMA_RLEN | R/W | 0x1 | DDMA read burst length<br>0: 8 data packages for one burst<br>1: 16 data packages for one burst<br>2, 3: Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [1:0] | DMA_WLEN | R/W | 0x1 | UDMA write burst length<br>0: 8 data packages for one burst<br>1: 16 data packages for one burst<br>2, 3: Reserved |

### 20.4.2.98 DMA FIFO Water Mark Register (Offset: 0x0204)

**Table 20-100. DMA FIFO Water Mark Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:20] | DSIF_DFIFO_WM | R/W | 0x2 | DSIF DDMA FIFO water mark<br>DDMA FIFO threshold for the reference of the DSIF operation<br>0: Don't care<br>1: 2/4 mem_src_width > FIFO filled data ≥ 1/4 mem_src_width<br>2: mem_src_width > FIFO filled data ≥ 2/4 mem_src_width<br>3: FIFO full |
| [19:18] | - | - | - | Reserved |
| [17:16] | DSIF_UFIFO_WM | R/W | 0x2 | DSIF UDMA FIFO water mark<br>UDMA FIFO threshold for the reference of the DSIF operation<br>0: Don't care<br>1: 1/4 mem_src_width ≤ FIFO left space < 2/4 mem_src_width<br>2: 2/4 mem_src_width ≤ FIFO left space<br>3: Reserved |
| [15: 2] | - | - | - | Reserved |
| [1:0] | UDMA_FIFO_HWM | R/W | 0 | UDMA FIFO high watermark for UDMA to alarm that FIFO is nearly full<br>0: FIFO left space < 1/8 FIFO size<br>1: 1/8 FIFO size ≤ FIFO left space < 2/8 FIFO size<br>2: 2/8 FIFO size ≤ FIFO left space < 3/8 FIFO size<br>3: Reserved |

### 20.4.2.99 Source Memory Base Address Register (Offset: 0x0210)

**Table 20-101. Source Memory Base Address Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_SRC_ST | R/W | 0 | Base address of the source image memory<br>DWORD alignment |

### 20.4.2.100 Source Memory Pitch Register (Offset: 0x0218)

**Table 20-102. Source Memory Pitch Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | MEM_SRC_PIT | R/W | 0 | Pitch of the source image for album mode |

### 20.4.2.101 Destination Memory Base Address 0 Register (Offset: 0x0220)

**Table 20-103. Destination Memory Base Address 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST0_YST | R/W | 0 | Destination base address of Y/YCbCr frame 0 image<br>DWORD alignment |

### 20.4.2.102 Destination Memory Base Address 1 Register (Offset: 0x0224)

**Table 20-104. Destination Memory Base Address 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST1_YST | R/W | 0 | Destination base address of Y/YCbCr frame 1 image<br>DWORD alignment |

### 20.4.2.103 Destination Memory Pitch Register (Offset: 0x0228)

**Table 20-105. Destination Memory Pitch Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | MEM_DEST_PIT | R/W | 0 | Pitch of the destination image for album mode |

### 20.4.2.104 Destination Memory Cb Base Address 0 Register (Offset: 0x0230)

**Table 20-106. Destination Memory Cb Base Address 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST0_CBST | R/W | 0 | Destination base address of Cb frame 0 image<br>DWORD alignment |

### 20.4.2.105 Destination Memory Cb Base Address 1 Register (Offset: 0x0234)

**Table 20-107. Destination Memory Cb Base Address 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST1_CBST | R/W | 0 | Destination base address of Cb frame 1 image<br>DWORD alignment |

### 20.4.2.106 Destination Memory Cr Base Address 0 Register (Offset: 0x0240)

**Table 20-108. Destination Memory Cr Base Address 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST0_CRST | R/W | 0 | Destination base address of Cr frame 0 image<br>DWORD alignment |

### 20.4.2.107 Destination Memory Cr Base Address 1 Register (Offset: 0x0244)

**Table 20-109. Destination Memory Cr Base Address 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST1_CRST | R/W | 0 | Destination base address of Cr frame 1 image.<br>DWORD alignment |

### 20.4.2.108 Command 0 Base Address Register (Offset: 0x0280)

**Table 20-110. Command 0 Base Address Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | CMD_BUF0_ST | R/W | 0 | Command buffer 0 start address<br>DWORD alignment |

### 20.4.2.109 Command X Base Address Register (Offset: 0x0284)

**Table 20-111. Command X Base Address Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | CMD_BUFX_ST | R/W | 0 | Command buffer X start address<br>DWORD alignment |

### 20.4.2.110 Command 1 Base Address Register (Offset: 0x0288)

**Table 20-112. Command 1 Base Address Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | CMD_BUF1_ST | R/W | 0 | Command buffer 1 start address<br>Special command buffer is used for loading parameters into Parameter RAM.<br>DWORD alignment |

### 20.4.2.111 AE Base Address 0 Register (Offset: 0x0290)

**Table 20-113. AE Base Address 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | AE_BUF0_ST | R/W | 0 | AE statistics destination buffer 0 base address<br>DWORD alignment |

### 20.4.2.112 AE Base Address 1 Register (Offset: 0x0294)

**Table 20-114. AE Base Address 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | AE_BUF1_ST | R/W | 0 | AE statistics destination buffer 1 base address<br>DWORD alignment |

### 20.4.2.113 AWB Base Address 0 Register (Offset: 0x0298)

**Table 20-115. AWB Base Address 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | AWB_BUF0_ST | R/W | 0 | AWB statistics destination buffer 0 base address<br>DWORD alignment |

### 20.4.2.114 AWB Base Address 1 Register (Offset: 0x029C)

**Table 20-116. AWB Base Address 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | AWB_BUF1_ST | R/W | 0 | AWB statistics destination buffer 1 base address<br>DWORD alignment |

### 20.4.2.115 AF Base Address 0 Register (Offset: 0x02A0)

**Table 20-117. AF Base Address 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | AF_BUF0_ST | R/W | 0 | AF statistics destination buffer 0 base address<br>DWORD alignment |

### 20.4.2.116 AF Base Address 1 Register (Offset: 0x02A4)

**Table 20-118. AF Base Address 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | AF_BUF1_ST | R/W | 0 | AF statistics destination buffer 1 base address<br>DWORD alignment |

### 20.4.2.117 HSTG Base Address Register (Offset: 0x02A8)

**Table 20-119. HSTG Base Address Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | HSTG_BUF_ST | R/W | 0 | Histogram statistics of the destination buffer base address<br>DWORD alignment |

### 20.4.2.118 MD Base Address Register (Offset: 0x02AC)

**Table 20-120. MD Base Address Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MD_BUF_ST | R/W | 0 | Motion detection statistics of the destination buffer base address<br>DWORD alignment |

### 20.4.2.119 Line Trigger Threshold Register (Offset: 0x0300)

**Table 20-121. Line Trigger Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:12] | - | - | - | Reserved |
| [11:0] | LN_TRIG_TH | R/W | 0 | Number of lines from the Sensor interface to trigger interrupt<br>Interrupt will be triggered if the specified number is reached. |

### 20.4.2.120 Sensor I/F Image Size Info Register (Offset: 0x03D0)

**Table 20-122. Sensor I/F Image Size Info Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | SIO_ACT_HEIGHT | R | 0 | Height of the SI output image<br>This information is only for reference. |
| [15:12] | - | - | - | Reserved |
| [11:0] | SIO_ACT_WIDTH | R | 0 | Width of the SI output image<br>This information is only for reference. |

### 20.4.2.121 Size down Image Size Info Register (Offset: 0x03D4)

**Table 20-123. Size-down Image Size Info Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | SDO_ACT_HEIGHT | R | 0 | Size-down the height of the output image<br>This information is only for reference. |
| [15:12] | - | - | - | Reserved |
| [11:0] | SDO_ACT_WIDTH | R | 0 | Size-down the width of the output image<br>This information is only for reference. |

### 20.4.2.122 Frame Counter Register (Offset: 0x03D8)

**Table 20-124. Frame Counter Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | FM_CNT_CLR | R/W | 0 | Write '1' to reset the frame counter (Auto clear) |
| [30:17] | - | - | - | Reserved |
| 16 | FM_CNT_SRC | R/W | 0 | Frame counter trigger source<br>0: Frame counter is triggered by Sensor I/F.<br>1: Frame counter is triggered by DMA frame done. |
| [15:0] | FM_CNT | R | - | Frame counter<br>Disable capture or set bit 31 will clear the frame counter. |

### 20.4.2.123 User Defined Register (Offset: 0x03DC)

**Table 20-125. User Defined Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | ISP_UDF | R/W | 0 | User defined register<br>User can program and read this register without affecting the ISP operation. |

### 20.4.2.124 STA Ready Status Register (Offset: 0x03E0)

**Table 20-126. STA Ready Status Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | MD_STA3_RDY | R | 0 | Motion detection window quarter 3 statistics is ready.<br>(Write one clear) |
| 30 | MD_STA2_RDY | R | 0 | Motion detection window quarter 2 statistics is ready.<br>(Write one clear) |
| 29 | MD_STA1_RDY | R | 0 | Motion detection window quarter 1 statistics is ready.<br>(Write one clear) |
| 28 | MD_STA0_RDY | R | 0 | Motion detection window quarter 0 statistics is ready.<br>(Write one clear) |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 27 | HSTG_STA_RDY | R | 0 | Histogram statistics is ready.<br>(Write one clear) |
| 26 | - | R | 0 | Reserved |
| 25 | AF_STA14_RDY | R | 0 | AF Window Row 4 statistics on the AF STA buffer 1 is ready.<br>AF STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 24 | AF_STA13_RDY | R | 0 | AF Window Row 3 statistics on the AF STA buffer 1 is ready.<br>AF STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 23 | AF_STA12_RDY | R | 0 | AF Window Row 2 statistics on the AF STA buffer 1 is ready.<br>AF STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 22 | AF_STA11_RDY | R | 0 | AF Window Row 1 statistics on the AF STA buffer 1 is ready.<br>AF STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 21 | AF_STA10_RDY | R | 0 | AF Window Row 0 statistics on AF STA buffer 1 is ready.<br>AF STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 20 | AF_STA04_RDY | R | 0 | AF Window Row 4 statistics on the AF STA buffer 0 is ready.<br>AF STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 19 | AF_STA03_RDY | R | 0 | AF Window Row 3 statistics on the AF STA buffer 0 is ready.<br>AF STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 18 | AF_STA02_RDY | R | 0 | AF Window Row 2 statistics on the AF STA buffer 0 is ready.<br>AF STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 17 | AF_STA01_RDY | R | 0 | AF Window Row 1 statistics on the AF STA buffer 0 is ready.<br>AF STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 16 | AF_STA00_RDY | R | 0 | AF Window Row 0 statistics on the AF STA buffer 0 is ready.<br>AF STA buffer 0 cannot store next STA data until the this bit is cleared.<br>(Write one clear) |
| 15 | AWB_STA13_RDY | R | 0 | AWB Window quarter 3 statistics on the STA buffer 1 is ready.<br>AWB STA buf 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 14 | AWB_STA12_RDY | R | 0 | AWB Window quarter 2 statistics on the STA buffer 1 is ready.<br>AWB STA buf 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 13 | AWB_STA11_RDY | R | 0 | AWB Window quarter 1 statistics on the STA buffer 1 is ready.<br>AWB STA buf 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 12 | AWB_STA10_RDY | R | 0 | AWB Window quarter 0 statistics on the STA buffer 1 is ready.<br>AWB STA buf 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 11 | AWB_STA03_RDY | R | 0 | AWB Window quarter 3 statistics on the STA buffer 0 is ready.<br>AWB STA buf 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 10 | AWB_STA02_RDY | R | 0 | AWB Window quarter 2 statistics on the STA buffer 0 is ready.<br>AWB STA buf 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 9 | AWB_STA01_RDY | R | 0 | AWB Window quarter 1 statistics on the STA buffer 0 is ready.<br>AWB STA buf 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 8 | AWB_STA00_RDY | R | 0 | AWB Window quarter 0 statistics on the STA buffer 0 is ready.<br>AWB STA buf 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 7 | AE_STA13_RDY | R | 0 | AE Window quarter 3 statistics on the STA buffer 1 is ready.<br>AE STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 6 | AE_STA12_RDY | R | 0 | AE Window quarter 2 statistics on the STA buffer 1 is ready.<br>AE STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 5 | AE_STA11_RDY | R | 0 | AE Window quarter 1 statistics on STA buffer 1 is ready.<br>AE STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 4 | AE_STA10_RDY | R | 0 | AE Window quarter 0 statistics on STA buffer 1 is ready.<br>AE STA buffer 1 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 3 | AE_STA03_RDY | R | 0 | AE Window quarter 3 statistics on STA buffer 0 is ready.<br>AE STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 2 | AE_STA02_RDY | R | 0 | AE Window quarter 2 statistics on STA buffer 0 is ready.<br>AE STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 1 | AE_STA01_RDY | R | 0 | AE Window quarter 1 statistics on STA buffer 0 is ready.<br>AE STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |
| 0 | AE_STA00_RDY | R | 0 | AE Window quarter 0 statistics on STA buffer 0 is ready.<br>AE STA buffer 0 cannot store next STA data until this bit is cleared.<br>(Write one clear) |

### 20.4.2.125 ISP Status Register (Offset: 0x03E4)

**Table 20-127. ISP Status Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | FD_STS | R | 0 | Frame done status<br>One frame has been uploaded completely to the main memory.<br>(Write One Clear) |
| 30 | FM_FAIL_ERR | R | 0 | Frame failed error<br>One frame cannot be processed completely until the start of the next frame.<br>(Write One Clear) |
| 29 | RAHB_IDLE_STS | R | 0 | AMBA AHB receives IDLE response at the Read direction<br>(Write One Clear) |
| 28 | WAHB_IDLE_STS | R | 0 | AMBA AHB receives IDLE response at the Write direction<br>(Write One Clear) |
| 27 | CMI_ERR | R | 0 | Command buffer error<br>(Write One Clear) |
| 26 | CMIX_END_STS | R | 0 | End of Command buffer X operation<br>(Write One Clear) |
| 25 | CMI1_END_STS | R | 0 | End of Command buffer 1 operation<br>(Write One Clear) |
| 24 | CMI0_END_STS | R | 0 | End of Command buffer 0 operation<br>(Write One Clear) |
| 23 | AHB_ERR | R | 0 | AMBA AHB receives error response<br>(Write One Clear) |
| 22 | VBLK_OP_ERR | R | 0 | CU or CMI operation cannot finish in the interval of the vertical blanking. The operation will complete but next frame will lose.<br>(Write One Clear) |
| 21 | DDMA_UDF_ERR | R | 0 | Down-stream DMA read underflow error<br>(Write One Clear) |
| 20 | UDMA_OVF_ERR | R | 0 | Up-stream DMA write overflow error<br>(Write One Clear) |
| 19 | LN_START_STS | R | 0 | Number of Sensor lines specified at LN_TRIG_TH register (0x300) has been received .<br>(Write One Clear) |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 18 | FM_END_STS | R | 0 | End of active frame status<br>(Write One Clear) |
| 17 | FM_START_STS | R | 0 | Start of the active frame status<br>(Write One Clear) |
| 16 | VSYNC_STS | R | 0 | VSYNC event on Sensor interface<br>(Write One Clear) |
| [15:13] | - | - | - | Reserved |
| 12 | GB_STA_INFO | R | 0 | Other statistics except the 3A DMA information<br>It indicates that MD STA cannot transfer one horizontal blanking line completely.<br>(Write One Clear). |
| [11:9] | - | - | - | Reserved |
| 8 | AF_STA_INFO | R | 0 | AF DMA information<br>It indicates that AF STA cannot transfer one horizontal blanking line completely.<br>(Write One Clear). |
| [7:5] | - | - | - | Reserved |
| 4 | AWB_STA_INFO | R | 0 | AWB DMA information<br>It indicates that AWB STA cannot transfer one horizontal blanking line completely.<br>(Write One Clear). |
| [3:1] | - | - | - | Reserved |
| 0 | AE_STA_INFO | R | 0 | AE DMA information. It indicates that AE STA cannot transfer one horizontal blanking line completely.<br>(Write One Clear) |

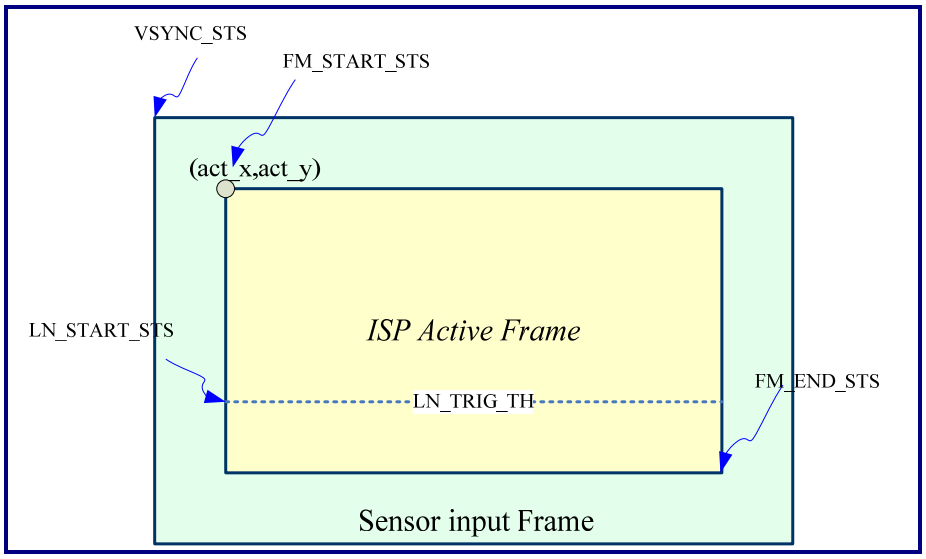



**Figure 20-5. Frame/Line Sync. Related Status**

## 20.4.2.126 ISP Interrupt Register (Offset: 0x03E8)

**Table 20-128. ISP Interrupt Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | FD_INT | R | 0 | Interrupt from fd_sts<br>(Write One Clear) |
| 30 | FM_FAIL_INT | R | 0 | Interrupt from fm_fail_err<br>(Write One Clear) |
| [29:28] | - | - | - | Reserved |
| 27 | CMI_ERR_INT | R | 0 | Interrupt from cmi_err<br>(Write One Clear) |
| 26 | CMIX_END_INT | R | 0 | Interrupt from cmix_end_sts<br>(Write One Clear) |
| 25 | CMI1_END_INT | R | 0 | Interrupt from cmi1_end_sts<br>(Write One Clear) |
| 24 | CMI0_END_INT | R | 0 | Interrupt from cmi0_end_sts<br>(Write One Clear) |
| 23 | AHB_ERR_INT | R | 0 | Interrupt from ahb_err<br>(Write One Clear) |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 22 | VBLK_OP_ERR_INT | R | 0 | Interrupt from vblk_op_err<br>(Write One Clear) |
| 21 | - | - | - | Reserved |
| 20 | UDMA_OVF_INT | R | 0 | Interrupt from udma_ovf_err<br>(Write One Clear) |
| 19 | LN_START_INT | R | 0 | Interrupt from ln_start_sts<br>(Write One Clear) |
| 18 | FM_END_INT | R | 0 | Interrupt from fm_end_sts<br>(Write One Clear) |
| 17 | FM_START_INT | R | 0 | Interrupt from fm_start_sts<br>(Write One Clear) |
| 16 | VSYNC_INT | R | 0 | Interrupt from vsync_sts<br>(Write One Clear) |
| 15 | STA_INFO_INT | R | 0 | Interrupt from ae_dma_info, awb_dma_info, af_dma_info, and gb_dma_info.<br>(Write One Clear) |
| [14:13] | - | - | 0 | Reserved |
| 12 | GB_STA_INT | R | 0 | Interrupt from statistics except 3A<br>(Write One Clear) |
| [11:9] | - | R | 0 | Reserved |
| 8 | AF_STA_INT | R | 0 | Interrupt from one of af_sta00_rdy ~ af_sta14_rdy<br>(Write One Clear) |
| [7:5] | - | R | 0 | Reserved |
| 4 | AWB_STA_INT | R | 0 | Interrupt from one of awb_sta00_rdy ~ awb_sta13_rdy<br>(Write One Clear) |
| [3:1] | - | R | 0 | Reserved |
| 0 | AE_STA_INT | R | 0 | Interrupt from one of ae_sta00_rdy ~ ae_sta13_rdy<br>(Write One Clear) |

### 20.4.2.127 ISP Interrupt Mask Register (Offset: 0x03EC)

**Table 20-129. ISP Interrupt Mask Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | FD_MASK | R/W | 0 | Frame done interrupt mask<br>0: Retain interrupt from fd_int<br>1: Prevent the occurrence of interrupt from fd_int |
| 30 | FM_FAIL_MASK | R/W | 0 | Frame fail interrupt mask<br>0: Retain interrupt from fm_fail_int<br>1: Prevent the occurrence of interrupt from fm_fail_int |
| [29:28] | - | - | - | Reserved |
| 27 | CMI_ERR_MASK | R/W | 0 | Command buffer Error interrupt mask<br>0: Retain interrupt from cmi_err_int<br>1: Prevent the occurrence of interrupt from cmi_err_int |
| 26 | CMIX_END_MASK | R/W | 0 | End of Command buffer x operation interrupt mask<br>0: Retain interrupt from cmix_end_int<br>1: Prevent the occurrence of interrupt from cmix_end_int |
| 25 | CMI1_END_MASK | R/W | 0 | End of Command buffer 1 operation interrupt mask<br>0: Retain interrupt from cmi1_end_int<br>1: Prevent the occurrence of interrupt from cmi1_end_int |
| 24 | CMI0_END_MASK | R/W | 0 | End of Command buffer 0 operation interrupt mask<br>0: Retain interrupt from cmi0_end_int<br>1: Prevent the occurrence of interrupt from cmi0_end_int |
| 23 | AHB_ERR_MASK | R/W | 0 | AHB error response interrupt mask<br>0: Retain interrupt from ahb_err_int<br>1: Prevent the occurrence of interrupt from ahb_err_int |
| 22 | VBLK_OP_ERR_MASK | R/W | 0 | Operation in Vertical Blanking Error interrupt mask<br>0: Retain interrupt source from vblk_op_err_int<br>1: Prevent the occurrence of interrupt from vblk_op_err_int |
| 21 | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 20 | UDMA_OVF_MASK | R/W | 0 | Up-stream DMA overflow interrupt mask<br>0: Retain interrupt from udma_ovf_int<br>1: Prevent the occurrence of interrupt from udma_ovf_int |
| 19 | LN_START_MASK | R/W | 1 | Line start interrupt mask<br>0: Retain interrupt from ln_start_int<br>1: Prevent the occurrence of interrupt from ln_start_int |
| 18 | FM_END_MASK | R/W | 1 | End of frame interrupt mask<br>0: Retain interrupt from fm_end_int<br>1: Prevent the occurrence of interrupt from fm_end_int |
| 17 | FM_START_MASK | R/W | 1 | Start of the frame interrupt mask<br>0: Retain interrupt from fm_start_int<br>1: Prevent the occurrence of interrupt from fm_start_int |
| 16 | VSYNC_MASK | R/W | 1 | VSYNC interrupt mask<br>0: Retain interrupt from vsync_int<br>1: Prevent the occurrence of interrupt from vsync_int |
| 15 | STA_INFO_MASK | R/W | 0 | STA DMA Info interrupt mask<br>0: Retain interrupt from stadma_info_int<br>1: Prevent the occurrence of interrupt from stadma_info_int |
| [14:13] | - | - | 0 | Reserved |
| 12 | GB_STA_MASK | R/W | 0 | Mask Interrupt from ISP statistics except 3A<br>0: Retain interrupt from gb_sta_int<br>1: Prevent the occurrence of interrupt from gb_sta_int |
| [11:9] | - | R/W | 0 | Reserved |
| 8 | AF_STA_MASK | R/W | 0 | AF statistics ready interrupt mask<br>0: Retain interrupt from af_sta_int<br>1: Prevent the occurrence of interrupt from af_sta_int |
| [7:5] | - | R/W | 0 | Reserved |
| 4 | AWB_STA_MASK | R/W | 0 | AWB statistics ready interrupt mask<br>0: Retain interrupt from awb_sta_int<br>1: Prevent the occurrence of interrupt from awb_sta_int |
| [3:1] | - | R/W | 0 | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 0 | AE_STA_MASK | R/W | 0 | AE statistics ready interrupt mask<br>0: Retain interrupt from ae_sta_int<br>1: Prevent the occurrence of interrupt from ae_sta_int |

### 20.4.2.128 ISP State Machine 0 Register (Offset: 0x03F0)

**Table 20-130. ISP State Machine 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:24] | CMI_CS | R | 0 | Command interpreter state machine |
| [23:22] | - | - | - | Reserved |
| [21:16] | MU_CU_CS | R | 0 | MU control CU state machine |
| [15:14] | - | - | - | Reserved |
| [13:8] | ISP_FM_CS | R | 0 | ISP Frame state machine |
| [7:6] | - | - | - | Reserved |
| [5:0] | ISP_MAIN_CS | R | 0 | ISP management unit Main state machine |

### 20.4.2.129 ISP State Machine 1 Register (Offset: 0x03F4)

**Table 20-131. ISP State Machine 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | CMIX_CNT | R | 0 | CMI X queue counter |
| [23:20] | - | - | - | Reserved |
| [19:16] | CMI0_CNT | R | 0 | CMI 0 queue counter |
| [15:14] | - | - | - | Reserved |
| [13:12] | DDMA_FIFO_LEV | R | 0 | DDMA FIFO filled level<br>0: 1/8 FIFO space > FIFO filled<br>1: 1/2 FIFO space > FIFO filled ≥ 1/8 FIFO space<br>2. 7/8 FIFO space > FIFO filled ≥ 1/2 FIFO space<br>3: FIFO filled ≥ 7/8 FIFO space |
| [11:10] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [9:8] | UDMA_FIFO_LEV | R | 0 | UDMA left FIFO space<br>0: left space ≥ 7/8 FIFO space<br>1: 7/8 FIFO space > left space ≥ 1/2 FIFO space<br>2: 1/2 FIFO space > left space ≥ 1/8 FIFO space<br>3: 1/8 FIFO space > left space |
| 7 | - | - | - | Reserved |
| [6:0] | UDMA_ARB_CS | R | 0 | UDMA Arbiter state machine |

### 20.4.2.130 ISP Test Pattern Register (Offset: 0x03F8)

**Table 20-132. ISP Test Pattern Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [29:24] | DT_TEST_PAT | R/W | 0 | Type of test pattern inserted before DT<br>[0] = 0: Disable the test pattern<br>[0] = 1: Enable the test pattern<br>[1] = 0: R = 0<br>[1] = 1: R = 255<br>[1] = 0: G = 0<br>[1] = 1: G = 255<br>[1] = 0: B = 0<br>[1] = 1: B = 255<br>[5:4] = 0: All pure color<br>[5:4] = 1: Coordinate RGB<br>[5:4] = 2: Vertical color bar<br>[5:4] = 3: Horizontal color bar |
| [23:14] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [13:8] | CI_TEST_PAT | R/W | 0 | Type of test pattern inserted before CI<br>[0] = 0: Disable the test pattern<br>[0] = 1: Enable the test pattern<br>[1] = 0: R = 0<br>[1] = 1: R = 255<br>[1] = 0: G = 0<br>[1] = 1: G = 255<br>[1] = 0: B = 0<br>[1] = 1: B = 255<br>[5:4] = 0: All pure color<br>[5:4] = 1: Coordinate RGB<br>[5:4] = 2: Vertical color bar<br>[5:4] = 3: Horizontal color bar |
| [7:6] | - | - | - | Reserved |
| [5:0] | SI_TEST_PAT | R/W | 0 | Type of test pattern inserted before SI<br>[0] = 0: Disable the test pattern<br>[0] = 1: Enable the test pattern<br>[1] = 0: R = 0<br>[1] = 1: R = 255<br>[1] = 0: G = 0<br>[1] = 1: G = 255<br>[1] = 0: B = 0<br>[1] = 1: B = 255<br>[5:4] = 0: All pure color<br>[5:4] = 1: Coordinate RGB<br>[5:4] = 2: Vertical color bar<br>[5:4] = 3: Horizontal color bar |

### 20.4.2.131 RDN Filter Threshold Register (Offset: 0x0400)

**Table 20-133. RDN Filter Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | RDN_TH4 | R/W | 0 | Threshold 4 of raw de-noise filter |
| [23:16] | RDN_TH3 | R/W | 0 | Threshold 3 of raw de-noise filter |
| [15:8] | RDN_TH2 | R/W | 0 | Threshold 3 of raw de-noise filter |
| [7:0] | RDN_TH1 | R/W | 0 | Threshold 3 of raw de-noise filter |

### 20.4.2.132 RDN Filter Weighting Register (Offset: 0x0404)

**Table 20-134. RDN Filter Weighting Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:19] | - | - | 0 | Reserved |
| [18:16] | RDN_THRAN | | | Raw de-noise delta threshold range |
| [15:12] | - | - | 0 | Reserved |
| [11:8] | RDN_W3 | R/W | 0 | Raw de-noise shift weighting 3 |
| [7:4] | RDN_W2 | R/W | 0 | Raw de-noise shift weighting 2 |
| [3:0] | RDN_W1 | R/W | 0 | Raw de-noise shift weighting 1 |

### 20.4.2.133 CDN Filter Threshold Register (Offset: 0x0408)

**Table 20-135. CDN Filter Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | CDN_TH4 | R/W | 0 | Threshold 4 of CI de-noise filter |
| [23:16] | CDN_TH3 | R/W | 0 | Threshold 3 of CI de-noise filter |
| [15:8] | CDN_TH2 | R/W | 0 | Threshold 3 of CI de-noise filter |
| [7:0] | CDN_TH1 | R/W | 0 | Threshold 3 of CI de-noise filter |

### 20.4.2.134 CDN Filter Weighting Register (Offset: 0x040C)

**Table 20-136. CDN Filter Weighting Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:19] | - | - | 0 | Reserved |
| [18:16] | CDN_THRAN | | | CI de-noise delta threshold range |
| [15:12] | - | - | 0 | Reserved |
| [11:8] | CDN_W3 | R/W | 0 | CI de-noise shift weighting 3 |
| [7:4] | CDN_W2 | R/W | 0 | CI de-noise shift weighting 2 |
| [3:0] | CDN_W1 | R/W | 0 | CI de-noise shift weighting 1 |

### 20.4.2.135 SP LPF Control Register (Offset: 0x0410)

**Table 20-137. SP LPF Control Threshold Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:5] | - | - | 0 | Reserved |
| 4 | SDN_LPF_SEL | R/W | 0 | SP de-noise LPF type select<br>0: Surrounding 8 pixels average<br>1: SP LPF |
| [3:2] | - | - | 0 | Reserved |
| [1:0] | SDN_LVL | R/W | 0 | 0: 25% LPF + 75% original pixel value<br>1: 50% LPF + 50% original pixel value<br>2: 75% LPF + 25% original pixel value<br>3: 100% LPF + 0% original pixel value |

### 20.4.2.136 SHDC RB Up Kx Register (Offset: 0x0440)

**Table 20-138. SHDC RB Up Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_B_KXU | R/W | 0 | Kx coefficient at the positive direction of y-axis for B shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [11:0] | SHDC_R_KXU | R/W | 0 | Kx coefficient at the positive direction of y-axis for R shading compensation<br>Format: 2.10 |

### 20.4.2.137 SHDC G Up Kx Register (Offset: 0x0444)

**Table 20-139. SHDC G Up Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_GB_KXU | R/W | 0 | Kx coefficient at the positive direction of y-axis for Gb shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |
| [11:0] | SHDC_GR_KXU | R/W | 0 | Kx coefficient at the positive direction of y-axis for Gr shading compensation<br>Format: 2.10 |

### 20.4.2.138 SHDC RB Down Kx Register (Offset: 0x0448)

**Table 20-140. SHDC RB Down Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_B_KXD | R/W | 0 | Kx coefficient at the negative direction of y-axis for B shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |
| [11:0] | SHDC_R_KXD | R/W | 0 | Kx coefficient at the negative direction of y-axis for R shading compensation<br>Format: 2.10 |

### 20.4.2.139 SHDC G Down Kx Register (Offset: 0x044C)

**Table 20-141. SHDC G Down Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_GB_KXD | R/W | 0 | Kx coefficient at the negative direction of y-axis for Gb shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |
| [11:0] | SHDC_GR_KXD | R/W | 0 | Kx coefficient at the negative direction of y-axis for Gr shading compensation<br>Format: 2.10 |

### 20.4.2.140 SHDC RB Right Kx Register (Offset: 0x0450)

**Table 20-142. SHDC RB Right Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_B_KXR | R/W | 0 | Kx coefficient at the positive direction of x-axis for B shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |
| [11:0] | SHDC_R_KXR | R/W | 0 | Kx coefficient at the positive direction of x-axis for R shading compensation<br>Format: 2.10 |

### 20.4.2.141 SHDC G Right Kx Register (Offset: 0x0454)

**Table 20-143. SHDC G Right Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_GB_KXR | R/W | 0 | Kx coefficient at the positive direction of x-axis for Gb shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [11:0] | SHDC_GR_KXR | R/W | 0 | Kx coefficient at the positive direction of x-axis for Gr shading compensation<br>Format: 2.10 |

### 20.4.2.142 SHDC RB Left Kx Register (Offset: 0x0458)

**Table 20-144. SHDC RB Left Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_B_KXL | R/W | 0 | Kx coefficient at the negative direction of x-axis for B shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |
| [11:0] | SHDC_R_KXL | R/W | 0 | Kx coefficient at the negative direction of x-axis for R shading compensation<br>Format: 2.10 |

### 20.4.2.143 SHDC G Left Kx Register (Offset: 0x045C)

**Table 20-145. SHDC G Left Kx Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | 0 | Reserved |
| [27:16] | SHDC_GB_KXL | R/W | 0 | Kx coefficient at the negative direction of x-axis for Gb shading compensation<br>Format: 2.10 |
| [15:12] | - | - | 0 | Reserved |
| [11:0] | SHDC_GR_KXL | R/W | 0 | Kx coefficient at the negative direction of x-axis for Gr shading compensation<br>Format: 2.10 |

### 20.4.2.144 SP Gain 0 Register (Offset: 0x0800)

**Table 20-146. SP Gain 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | - | - | 0 | Reserved |
| [30:24] | SP_K3 | R/W | 0 | Sharpness gain of segment 3<br>Format: 3.4 |
| 23 | - | - | 0 | Reserved |
| [22:16] | SP_K2 | R/W | 0 | Sharpness gain of segment 2<br>Format: 3.4 |
| 15 | - | - | 0 | Reserved |
| [14:8] | SP_K1 | R/W | 0 | Sharpness gain of segment 1<br>Format: 3.4 |
| 7 | - | - | 0 | Reserved |
| [6:0] | SP_K0 | R/W | 0 | Sharpness gain of segment 0<br>Format: 3.4 |

### 20.4.2.145 SP Gain 1 Register (Offset: 0x0804)

**Table 20-147. SP Gain 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | SP_K7 ~ SP_K4 | R/W | 0 | Sharpness gain of segments 4 ~ 7<br>Format: 3.4<br>Please refer to SP Gain 0 for more details. |

### 20.4.2.146 SP Gain 2 Register (Offset: 0x0808)

**Table 20-148. SP Gain 2 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | SP_K11 ~ SP_K8 | R/W | 0 | Sharpness gain of segments 8 ~ 11<br>Format: 3.4<br>Please refer to SP Gain 0 for more details. |

### 20.4.2.147 SP Gain 3 Register (Offset: 0x080C)

**Table 20-149. SP Gain 3 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | SP_K15 ~ SP_K12 | R/W | 0 | Sharpness gain of segments 12 ~ 15<br>Format: 3.4<br>Please refer to SP Gain 0 for more details. |

### 20.4.2.148 SP Gain 4 Register (Offset: 0x0810)

**Table 20-150. SP Gain 4 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | SP_K19~SP_K16 | R/W | 0 | Sharpness gain of segments 16 ~ 19<br>Format: 3.4<br>Please refer to SP Gain 0 for more details. |

### 20.4.2.149 SP Weighting 0 Register (Offset: 0x0820)

**Table 20-151. SP Weighting 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | 0 | Reserved |
| [28:24] | SP_W3 | R/W | 0 | Sharpness weight ration of segment 3<br>Format: 1.4 |
| [23:21] | - | - | 0 | Reserved |
| [20:16] | SP_W2 | R/W | 0 | Sharpness weight ratio of segment 2<br>Format: 1.4 |
| [15:13] | - | - | 0 | Reserved |
| [12:8] | SP_W1 | R/W | 0 | Sharpness weight ratio of segment 1<br>Format: 1.4 |
| [7:5] | - | - | 0 | Reserved |
| [4:0] | SP_W0 | R/W | 0 | Sharpness weight ratio of segment 0<br>Format: 1.4 |

### 20.4.2.150 SP Weighting N Register (Offset: 0x0824 ~ 0x083F)

**Table 20-152. SP Weighting N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | SP_W7 ~ SP_W4 | R/W | 0 | Sharpness weight ratios of segments 4 ~ 7<br>Format: 1.4<br>Please refer to SP Weighting 0 for more details. |
| [31:0] | SP_W11 ~ SP_W8 | R/W | 0 | Sharpness weight ratios of segments 8 ~ 11<br>Format: 1.4 |
| [31:0] | SP_W15 ~ SP_W12 | R/W | 0 | Sharpness weight ratios of segments 12 ~ 15<br>Format: 1.4 |
| [31:0] | SP_W19 ~ SP_W16 | R/W | 0 | Sharpness weight ratios of segments 16 ~ 19<br>Format: 1.4 |
| [31:0] | SP_W23 ~ SP_W20 | R/W | 0 | Sharpness weight ratios of segments 20 ~ 23<br>Format: 1.4 |
| [31:0] | SP_W27 ~ SP_W24 | R/W | 0 | Sharpness weight ratios of segments 24 ~ 27<br>Format: 1.4 |
| [31:0] | SP_W31 ~ SP_W28 | R/W | 0 | Sharpness weight ratios of segments 28 ~ 31<br>Format: 1.4 |

### 20.4.2.151 Gamma R LUT 0 Register (Offset: 0x1000)

**Table 20-153. Gamma R LUT 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:20] | - | - | 0 | Reserved |
| [19:10] | GM_RLUT_1 | R/W | 0 | Mapping valu of Gamma R LUT 1<br>Format: 10 |
| [9:0] | GM_RLUT_0 | R/W | 0 | Mapping value of Gamma R LUT R 0<br>Format: 10 |

### 20.4.2.152 Gamma R LUT N1 Register (Offset: 0x1004 ~ 0x1FFF)

**Table 20-154. Gamma R LUT N1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | GM_RLUT_3 ~ GM_RLUT_2 | R/W | 0 | Mapping values of Gamma R LUT 2 ~ 3<br>Format: 10 |
| [31:0] | GM_RLUT_5 ~ GM_RLUT_4 | R/W | 0 | Mapping values of Gamma R LUT R 4 ~ 5<br>Format: 10 |
| [31:0] | GM_RLUT_2047 ~ GM_RLUT_2046 | R/W | 0 | Mapping values of Gamma R LUT R 2046 ~ 2047<br>Format: 10 |

### 20.4.2.153 Gamma R LUT 2048 Register (Offset: 0x2000)

**Table 20-155. Gamma R LUT 2048 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:20] | - | - | 0 | Reserved |
| [19:10] | GM_RLUT_2050_1 | R/W | 0 | Mapping values of Gamma R LUT R 2050 and 2051<br>Format: 10 |
| [9:0] | GM_RLUT_2048_9 | R/W | 0 | Mapping values of Gamma R LUT 2048 and 2049<br>Format: 10 |

### 20.4.2.154 Gamma R LUT N2 Register (Offset: 0x2004 ~ 0x23FF)

**Table 20-156. Gamma R LUT N2 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | GM_RLUT_2055 ~ GM_RLUT_2052 | R/W | 0 | Mapping values of Gamma R LUT 2052 ~ 2055<br>Format: 10 |
| [31:0] | GM_RLUT_2059 ~ GM_RLUT_2056 | R/W | 0 | Mapping values of Gamma R LUT 2056 ~ 2059<br>Format: 10 |
| [31:0] | GM_RLUT_3071 ~ GM_RLUT_3068 | R/W | 0 | Mapping values of Gamma R LUT 3068 ~ 3071<br>Format: 10 |

### 20.4.2.155 Gamma R LUT 3072 Register (Offset: 0x2400)

**Table 20-157. Gamma R LUT 3072 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:20] | - | - | 0 | Reserved |
| [19:10] | GM_RLUT_3076_9 | R/W | 0 | Mapping values of Gamma R LUT 3076 ~ 3079<br>Format: 10 |
| [9:0] | GM_RLUT_3072_5 | R/W | 0 | Mapping values of Gamma R LUT R 3072 ~ 3075<br>Format: 10 |

### 20.4.2.156 Gamma R LUT N3 Register (Offset: 0x2404 ~ 0x25FF)

**Table 20-158. Gamma R LUT N3 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | GM_RLUT_3087 ~ GM_RLUT_3080 | R/W | 0 | Mapping values of Gamma R LUT 3080 ~ 3087<br>Format: 10 |
| [31:0] | GM_RLUT_3095 ~ GM_RLUT_3088 | R/W | 0 | Mapping values of Gamma R LUT 3088 ~ 3095<br>Format: 10 |
| [31:0] | GM_RLUT_4095 ~ GM_RLUT_4088 | R/W | 0 | Mapping values of Gamma R LUT 4088 ~ 4095<br>Format: 10 |

### 20.4.2.157 Gamma G LUT Register (Offset: 0x3000 ~ 0x45FF)

**Table 20-159. Gamma G LUT Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | GM_GLUT_0 ~ GM_GLUT_4095 | R/W | 0 | Mapping values of Gamma G LUT 0 ~ 4095<br>Format: 10<br>Please refer to Gamma R LUT for more details. |

### 20.4.2.158 Gamma B LUT Register (Offset: 0x5000 ~ 0x65FF)

**Table 20-160. Gamma B LUT Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:0] | GM_BLUT_0 ~ GM_BLUT_4095 | R/W | 0 | Mapping values of Gamma B LUT 0 ~ 4095<br>Format: 10<br>Please refer to Gamma R LUT for more details. |

### 20.4.2.159 Raw R Mapping Segment 0 Register (Offset: 0x7000)

**Table 20-161. Raw R Mapping Segment 0 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_R_SLP0 | R/W | 0 | Slop of Raw R mapping segment 0<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | - | - | - | Reserved |

### 20.4.2.160 Raw R Mapping Segment 1 Register (Offset: 0x7004)

**Table 20-162. Raw R Mapping Segment 1 Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_R_SLP1 | R/W | 0 | Slop of raw R mapping segment 1<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | RAW_R_YPT1 | R/W | 0 | Output start point of raw R segment 1<br>Format: 12 |

### 20.4.2.161 Raw R Mapping Segment N Register (Offset: 0x7008 ~ 0x73FF)

**Table 20-163. Raw R Mapping Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_R_SLP2 ~ RAW_R_SLP255 | R/W | 0 | Slops of raw R segments 2 ~ 255<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | RAW_R_YPT2 ~ RAW_R_YPT255 | R/W | 0 | Output start points of raw R segments 2 ~ 255<br>Format: 12 |

### 20.4.2.162 Raw B Mapping Segment N Register (Offset: 0x7400 ~ 0x77FF)

**Table 20-164. Raw B Mapping Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_B_SLP0 ~ RAW_B_SLP255 | R/W | 0 | Slops of raw B segments 0 ~ 255<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | RAW_B_YPT1 ~ RAW_B_YPT255 | R/W | 0 | Output stat points of raw R segments 1 ~ 255<br>Format: 12 |

### 20.4.2.163 Raw Gr Mapping Segment N Register (Offset: 0x8000 ~ 0x8BFF)

**Table 20-165. Raw Gr Mapping Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_Gr_SLP0 ~ RAW_Gr_SLP255 | R/W | 0 | Slops of raw Gr segments 0 ~ 255<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | RAW_Gr_YPT1 ~ RAW_Gr_YPT255 | R/W | 0 | Output stat points of raw Gr segments 1 ~ 255<br>Format: 12 |

### 20.4.2.164 Raw Gb Mapping Segment N Register (Offset: 8C00 ~ 8FFF)

**Table 20-166. Raw Gb Mapping Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:16] | RAW_Gb_SLP0 ~ RAW_Gb_SLP255 | R/W | 0 | Slops of raw Gb segments 0 ~ 255<br>Format: 2.10 |
| [15:12] | - | - | - | Reserved |
| [11:0] | RAW_Gb_YPT1 ~ RAW_Gb_YPT255 | R/W | 0 | Output start points of raw Gb segments 1 ~ 255<br>Format: 12 |

### 20.4.2.165 YCC Luma Curve Segment N Register (Offset: 2800 ~ 29FF)

**Table 20-167. YCC Luma Curve Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | YCC_Y_SLP0 ~ YCC_Y_SLP127 | R/W | 0 | Slops of YCC Luma curve segments 0 ~ 127<br>Format: 4.5 |
| [15:8] | - | - | - | Reserved |
| [7:0] | YCC_Y_YPT1 ~ YCC_Y_YPT127 | R/W | 0 | Output star points of YCC Luma curve segments 1 ~ 127<br>Format: 8 |

### 20.4.2.166 YCC Chroma Curve Segment N Register (Offset: 2C00 ~ 2CFF)

**Table 20-168. YCC Chroma Curve Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | YCC_C_SLP0 ~ YCC_C_SLP127 | R/W | 0 | Slops of YCC Luma curve segments 0 ~ 63<br>Format: 3.4 |
| [15:7] | - | - | - | Reserved |
| [6:0] | YCC_C_YPT1 ~ YCC_C_YPT127 | R/W | 0 | Output start points of YCC Luma curve segments 1 ~ 63<br>Format: 7 |

### 20.4.2.167 CS Gain Curve Segment N Register (Offset: 4800 ~ 48FF)

**Table 20-169. CS Gain Curve Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | CS_GSLP_SIGN0 | - | - | Slop sign of CS gain segment 0 |
| [30:26] | - | - | - | Reserved |
| [25:16] | CS_GSLP0 ~ CS_GSLP63 | R/W | 0 | Slops of CS gain segments 0 ~ 63<br>Format: 10 |
| [15:7] | - | - | - | Reserved |
| [6:0] | CS_GYPTR1 ~ CS_GYPTR63 | R/W | 0 | Output start points of CS gain segments 1 ~ 63<br>Format: 1.6 |

### 20.4.2.168 CS Weighting Segment N Register (Offset: 4900 ~ 493F)

**Table 20-170. CS Weighting Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:24] | CS_WYPTR3,<br>CS_WYPTR7,<br>……,<br>CS_GYPTR63 | R/W | 0 | Output points of CS weighting segments 3, 7, ... 63<br>Format: 1.4 |
| [23:21] | - | - | - | Reserved |
| [20:16] | CS_WYPTR2,<br>CS_WYPTR6,<br>……,<br>CS_GYPTR62 | R/W | 0 | Output points of CS weighting segments 2, 6, ... 62<br>Format: 1.4 |
| [15:13] | - | - | - | Reserved |
| [12:8] | CS_WYPTR1,<br>CS_WYPTR5,<br>……,<br>CS_GYPTR61 | R/W | 0 | Output points of CS weighting segments 1, 5, ... 61<br>Format: 1.4 |
| [5:7] | - | - | - | Reserved |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [4:0] | CS_WYPTR0,<br>CS_WYPTR4,<br>……,<br>CS_GYPTR60 | R/W | 0 | Output points of CS weighting segments 0, 4, ... 60<br>Format: 1.4 |

### 20.4.2.169 BCS Gain Curve Segment N Register (Offset: 4F00 ~ 4F7F)

**Table 20-171. BCS Gain Curve Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| 31 | CS_GSLP_SIGN0 | - | - | Slop sign of BCS gain segment 0 |
| [30:26] | - | - | - | Reserved |
| [25:16] | CS_GSLP0 ~ CS_GSLP31 | R/W | 0 | Slops of BCS gain segments 0 ~ 63<br>Format: 10 |
| [15:7] | - | - | - | Reserved |
| [6:0] | CS_GYPTR1 ~ CS_GYPTR31 | R/W | 0 | Output start points of BCS gain segments 1 ~ 63<br>Format: 1.6 |

### 20.4.2.170 IE Saturation Gain Segment N Register (Offset: 4A00 ~ 4A7F)

**Table 20-172. IE Saturation Gain Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | IE_SATG 3,<br>IE_SATG 7,<br>……,<br>IE_SATG 127 | R/W | 0 | Saturation gain control for segments 3, 7, ..., 127<br>Format: 3.5 |
| [23:16] | IE_SATG 2,<br>IE_SATG 6,<br>……,<br>IE_SATG 126 | R/W | 0 | Saturation gain control for Segment 2, 6..., 126<br>Format: 3.5 |
| [15:8] | IE_SATG 1,<br>IE_SATG 5,<br>……,<br>IE_SATG 125 | R/W | 0 | Saturation gain control for segments 1, 5, ..., 125<br>Format: 3.5 |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [7:0] | IE_SATG 0,<br>IE_SATG 4,<br>……,<br>IE_SATG 124 | R/W | 0 | Saturation gain control for segments 0, 4, ..., 124<br>Format: 3.5 |

### 20.4.2.171 IE Saturation Hue Segment N Register (Offset: 4B00 ~ 4B58)

**Table 20-173. IE Saturation Hue Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:24] | IE_SATH 3,<br>IE_SATH 7,<br>……,<br>IE_SATH 87 | R/W | 0 | Hue ratio of saturation gain for segments 3, 7, ..., 87<br>Format: .6 |
| [23:22] | - | - | - | Reserved |
| [21:16] | IE_SATH 2,<br>IE_SATH 6,<br>……,<br>IE_SATH 86 | R/W | 0 | Hue ratio of saturation gain for segments 2, 6, ..., 86<br>Format: .6 |
| [15:14] | - | - | - | Reserved |
| [13: 8] | IE_SATH 1,<br>IE_SATH 5,<br>……,<br>IE_SATH 89 | R/W | 0 | Hue ratio of saturation gain for segments 1, 5, ..., 89<br>Format: .6 |
| [7:6] | - | - | - | Reserved |
| [5:0] | IE_SATH 0,<br>IE_SATH 4,<br>……,<br>IE_SATH 88 | R/W | 0 | Hue ratio of saturation gain for segments 0, 4, ..., 88<br>Format: .6 |

### 20.4.2.172 IE Saturation to Y Offset Segment N Register (Offset: 4C00 ~ 4C3F)

**Table 20-174. IE Saturation to Y Offset Segment N Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | IE_SATY 3,<br>IE_SATY 7,<br>……,<br>IE_SATY 63 | R/W | 0 | Saturation gain of Y offset for segments 3, 7, ..., 127<br>Format: .8<br>Segments 0 ~ 7: Max. = 0x1F<br>Segments 8 ~ 15: Max. = 0x3F<br>Segments 16 ~ 23: Max. = 0x7F |
| [23:16] | IE_SATY 2,<br>IE_SATY 6,<br>……,<br>IE_SATY 62 | R/W | 0 | Saturation gain of Y offset for segments 2, 6, ..., 126<br>Format: .8<br>Segments 0 ~ 7: Max. = 0x1F<br>Segments 8 ~ 15: Max. = 0x3F<br>Segments 16 ~ 23: Max. = 0x7F |
| [15:8] | IE_SATY 1,<br>IE_SATY 5,<br>……,<br>IE_SATY 61 | R/W | 0 | Saturation gain of Y offset for segments 1, 5, ..., 125<br>Format: .8<br>Segments 0 ~ 7: Max. = 0x1F<br>Segments 8 ~ 15: Max. = 0x3F<br>Segments 16 ~ 23: Max. = 0x7F |
| [7:0] | IE_SATY 0,<br>IE_SATY 4,<br>……,<br>IE_SATY 60 | R/W | 0 | Saturation gain of Y offset for segments 0, 4, ..., 124<br>Format: .8<br>Segments 0 ~ 7: Max. = 0x1F<br>Segments 8 ~ 15: Max. = 0x3F<br>Segments 16 ~ 23: Max. = 0x7F |

### 20.4.2.173 Hue Rotation θ Table Register (Offset: 4D00 ~ 4D58)

**Table 20-175. Hue Rotation θ Table Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:24] | HUE_R12_15,<br>HUE_R28_31,<br>……,<br>HUR_R348_351 | R/W | 0 | Hue rotatation θ for 12° ~ 15°, 28° ~ 31°, …, 348° ~ 351°<br>Format: 8 |

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [23:16] | HUE_R8_11,<br>HUE_R24_27,<br>……,<br>HUR_R344_347 | R/W | 0 | Hue rotation θ for 8° ~ 11°, 24° ~ 27°, …, 344° ~ 347°<br>Format: 8 |
| [15:8] | HUE_R4_7,<br>HUE_R20_23,<br>……,<br>HUR_R356_359 | R/W | 0 | Hue rotation θ for 4° ~ 7°, 20° ~ 23°, …, 356° ~ 359°<br>Format: 8 |
| [7:0] | HUE_R0_3,<br>HUE_R16_19,<br>……,<br>HUR_R352_355 | R/W | 0 | Hue rotation θ for 0° ~ 3°, 16° ~ 19°, …, 352° ~ 355°<br>Format: 8 |

### 20.4.2.174 CE Fine Tuning Gain Register (Offset: 4E00 ~ 4E1F)

**Table 20-176. CE Fine Tuning Gain Register**

| Bit | Name | Type | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | CE_G1,<br>CE_G3,<br>……,<br>CE_G61 | R/W | 0 | CE fine tuning gain for segments 1, 3, ..., 61<br>Format: 2.8 |
| [15:11] | - | - | - | Reserved |
| [10:0] | CE_G0,<br>CE_G2,<br>……,<br>CE_G62 | R/W | 0 | CE fine tuning gain for segments 0, 2, ..., 62<br>Format: 2.8 |

## 20.5 Block Descriptions

### 20.5.1 Sensor I/F

The protocol of the sensor I/F is controlled by the ISP configuration the SVERF_DEF and SHREF_DEF (0x14) register. If this register is set to '1', the SYNC I/F mode will be used; otherwise, the DATA VALID I/F mode will be used. Users can mix these modes on the vertical and horizontal control signals.

#### 20.5.1.1 SVREF_DEF = 1 and SHREF_DEF = 1

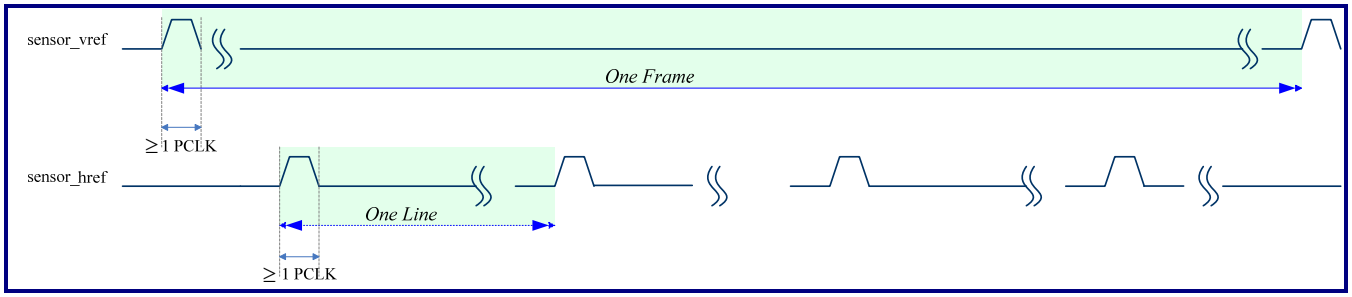



**Figure 20-6. SVREF_DEF = 1 and SHREF_DEF = 1**

#### 20.5.1.2 SVREF_DEF = 0 and SHREF_DEF = 0

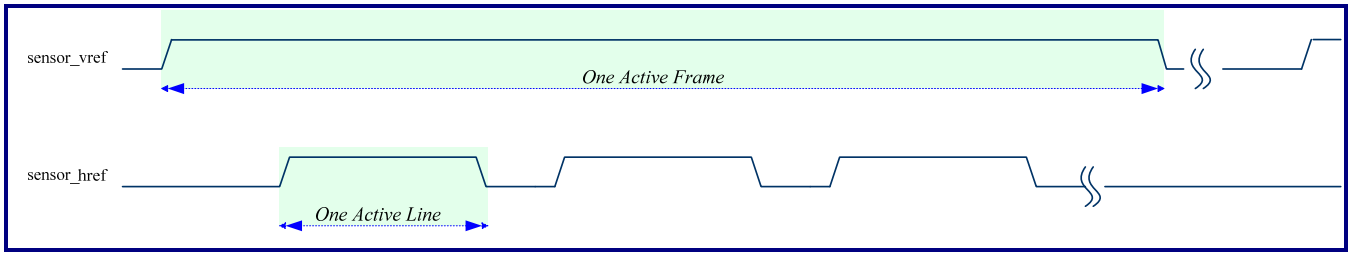



**Figure 20-7. SVREF_DEF = 0 and SHREF_DEF = 0**

#### 20.5.1.3 SVREF_DEF = 1 and SHREF_DEF = 0

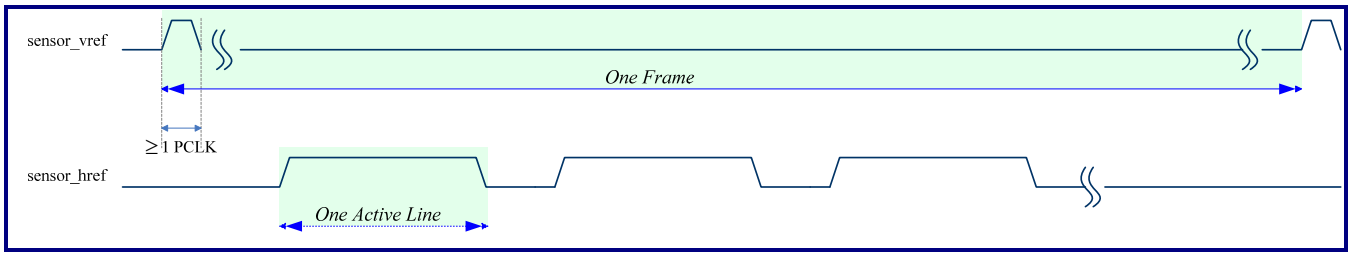



**Figure 20-8. SVREF_DEF = 1 and SHREF_DEF = 0**

### 20.5.2 Sensor Correction

The sensor correction circuit includes the following blocks.

- Black Level Calibration removes the sensor optical black value.
- Raw Mapping Curve on R/Gr/Gb/B raw data liberalizes the sensor response.
- Shading Correction compensates the lens shading effect.
- Defect Pixel Correction corrects the bad pixel.
- Gb/Gr Crosstalk Removal compensates the sensor different response between Gb and Gr cells.

### 20.5.3 RGB Gain

Pre-Gain works on the Raw-data domain and the RGB gain after Color Interpolation achieves near 16X total gain.

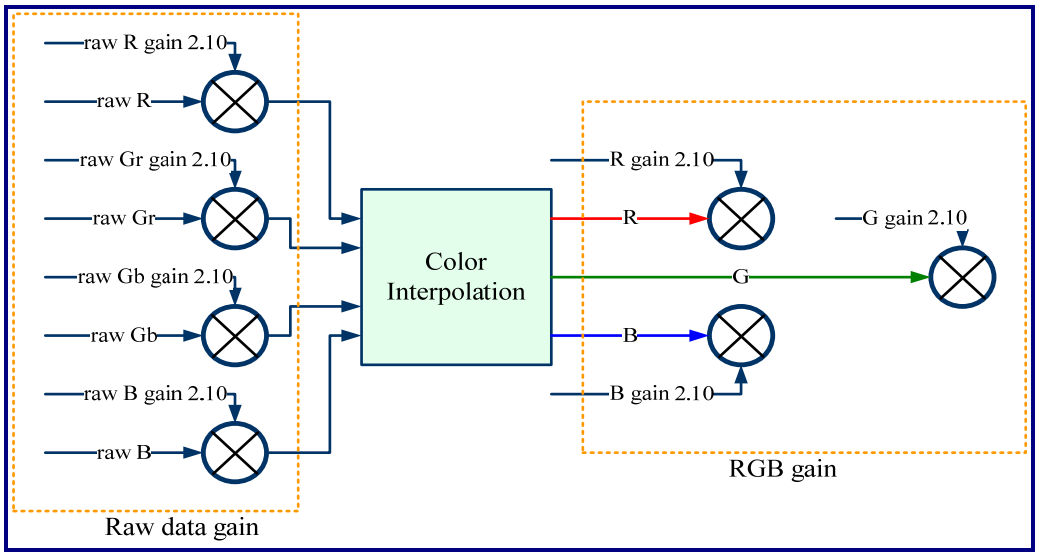



**Figure 20-9. RGB Gain**

### 20.5.4 Resize

The size can be changed easily by programming the SENSOR_ACT_WIDTH & SENSOR_ACT_HEIGHT (0x28) register and the TARGET_WIDTH and TARGET_HEIGHT (0x30) register. The aspect ratio can be kept depending on the ASPECT_MAJOR (0x0C[21:20]) register.

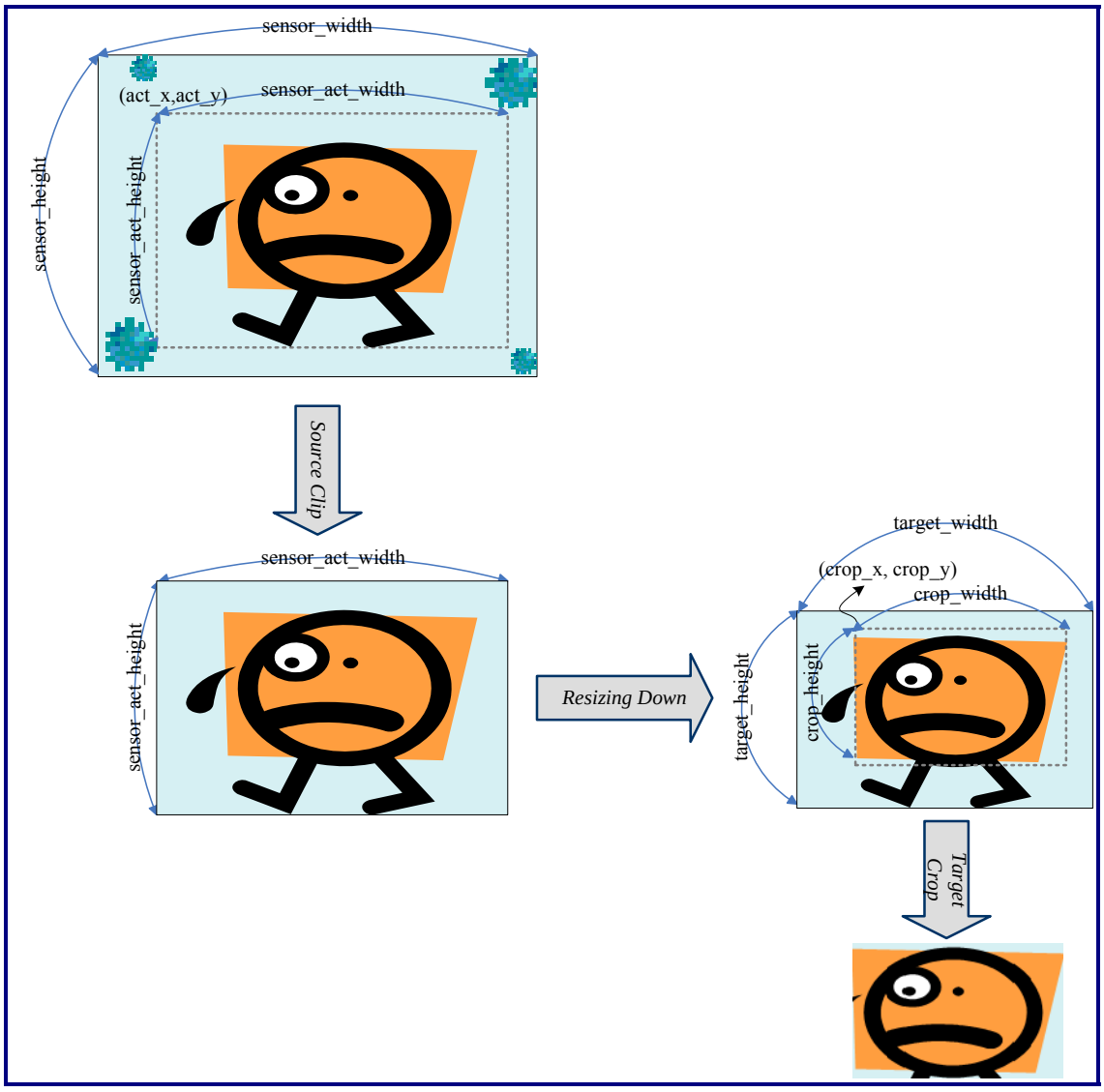



**Figure 20-10. Resizing Down**

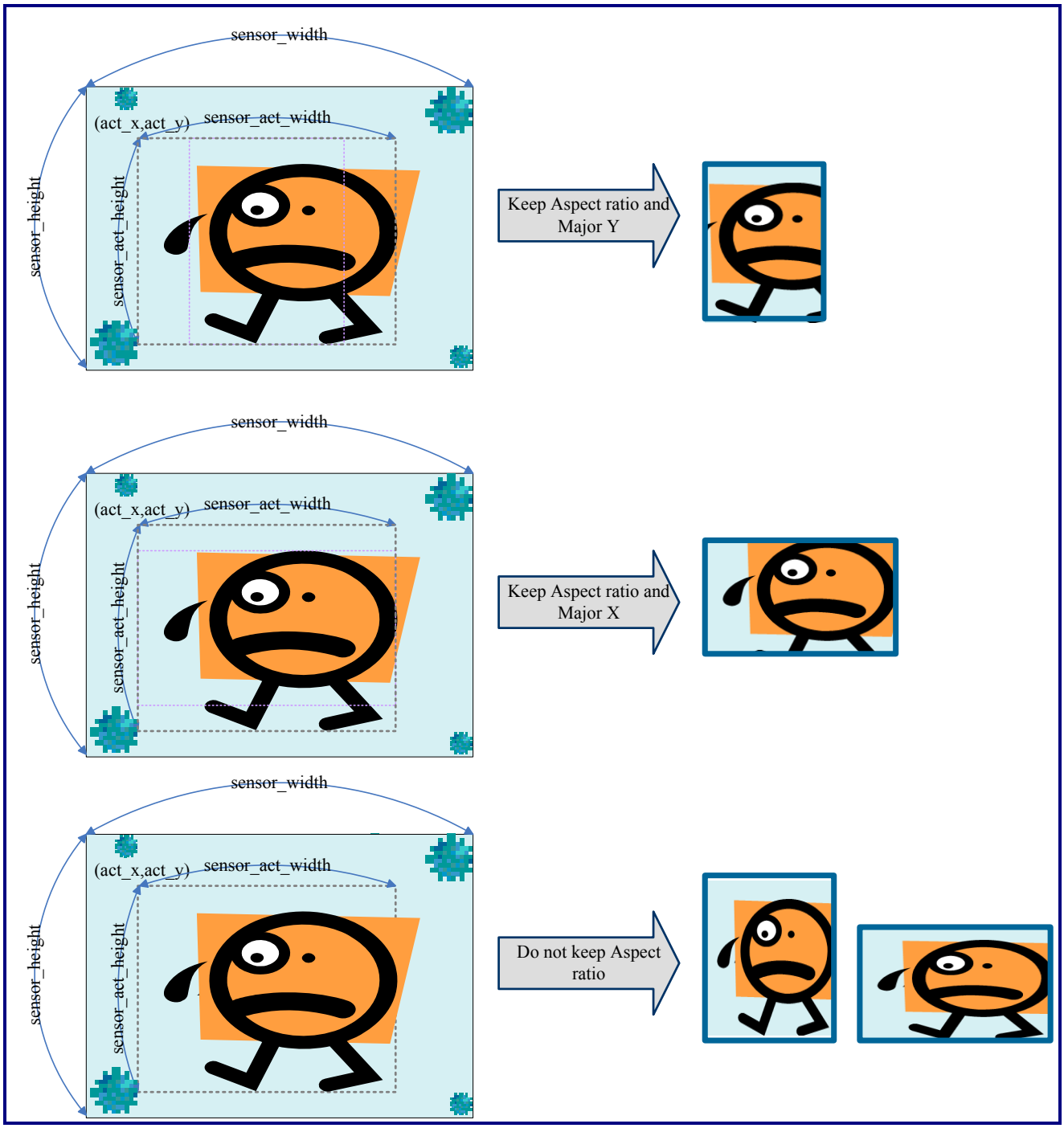



**Figure 20-11. Aspect Ratio on Sizing Down**

## 20.6 Statistics of 3A, Histogram, and Motion Detection

Users can obtain the AE, AWB, and MD statistics at a quarter of the AE/AWB/MD window, or at the end of a frame. Users can obtain the AF statistics according to the AF window row or at the end of a frame. The frame number will be attached to the 3A statistics. The destination memory of the 3A statistic is a pin-pong buffer or a single buffer. The overwrite mode and the non-overwrite mode are provided for the convenience of user access. In the overwrite mode, the 3A statistic will write to the STA buffer unconditionally. In the non-overwrite mode, the 3A statistic will not write to the STA buffer until the RDY bits of related register are cleared. It is best to set the start point of the AE window to an odd number and the AWB window to an even number.

## 20.7 Register and Table Access

One of the following methods should be used to program the Configuration register and LUT.

1. CPU reads/writes register/LUT by the AHB slave directly.
2. CPU triggers ISP HW to dump a value into register/LUT automatically. Depending on the user trigger setting (Offset: 0x04), ISP HW will launch the programming action at any time or during the frame vertical blanking interval.

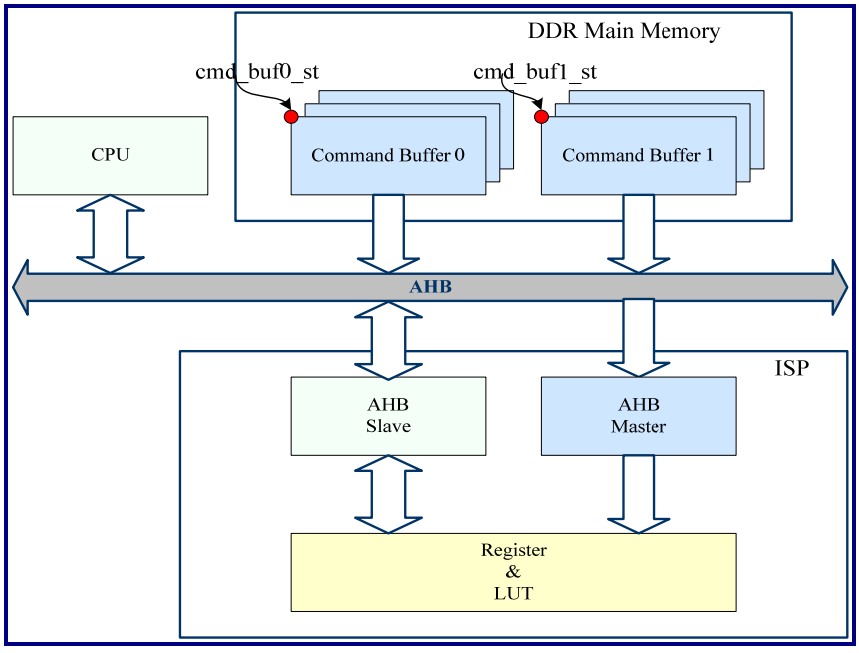



**Figure 20-12. Register and Table Access**

# Chapter 21

# NAND-type Flash Controller

This chapter contains the following sections:


- 21.1 General Description
- 21.2 Features
- 21.3 Block Diagram
- 21.4 Summary of Control Registers
- 21.5 Register Definitions
- 21.6 AHB Data Slave Port
- 21.7 Buffer Management Controller
- 21.8 AHB Register Slave Port
- 21.9 ECC Correction Error Handling
- 21.10 Auto Compare Error Handling
- 21.11 Abort Sequence
- 21.12 Rom/SRAM Behavior
- 21.13 Host Controller Data Format
- 21.14 Description of NANDC MicroCode

## 21.1 General Description

The NAND Flash controller will act as an interface to access the NAND-type Flash. The hardware-based NAND host controller allows data to be transferred in the high-speed mode. The ECC information is automatically generated in the programming operation and checked in the read operation.

## 21.2 Features

- Supports 2 K/4 K/8 K page sizes of NAND Flash
- Supports 8-bit data bus NAND Flash
- Each channel can connect up to 4 CEs (Flash chip enable)
- Supports programmable NAND Flash timing parameters
- Supports programmable NAND Flash command flow
- Supports interleaving operation when two or more CEs connected to one channel
- Supports CRC generation/check by hardware while accessing user-defined data
- Synchronous/Asynchronous clock mode between core and Flash memory interface
- Supports up to 24-bit ECC correction capability of a 512 or 1 K byte sector
- Supports DMA handshake mode
- Provides AHB data slave port to support RETRY response if the DMA handshake mode is not exploited

## 21.3 Block Diagram

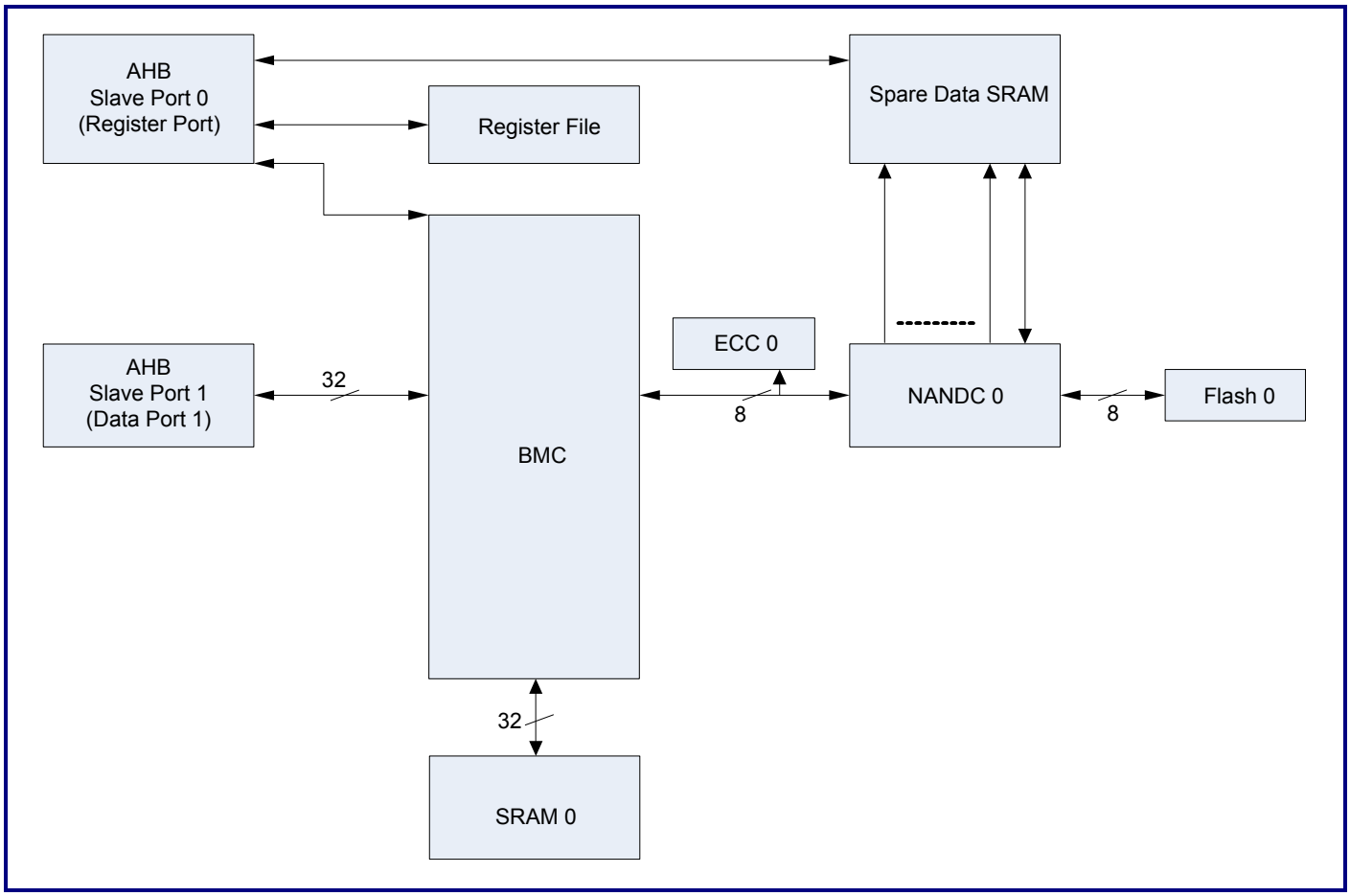



**Figure 21-1. Block Diagram**

## 21.4 Summary of Control Registers

The NAND Flash control registers are summarized in Table 21-1. The register between the offsets 0x0 to 0x204 and 0x400 to 0x1FFFF can be accessed with BYTE, HWORD, or WORD as HSIZE. The command queue input entry can only accept WORD as HSIZE.

**Table 21-1. Summary of Control Registers**

| Address Offset | Type | Description | Reset Value |
|---|---|---|---|
| **ECC Control** | | | |
| 0x0_0000 | R | ECC Status Register 0 | 0x0000_0000 |
| 0x0_0004 | R | ECC Status Register 1 | 0x0000_0000 |
| 0x0_0008 | R/W | ECC Control Register | 0x0000_0000 |
| 0x0_0010 | R/W | Threshold Number of ECC Error Bits Register 0 | 0x0000_0000 |
| 0x0_0014 | R/W | Threshold Number of ECC Error Bits Register 1 | 0x0000_0000 |
| 0x0_0018 | R/W | Number of ECC Correction Capability Bits Register 0 | 0x0000_0000 |
| 0x0_001C | R/W | Number of ECC Correction Capability Bits Register 1 | 0x0000_0000 |
| 0x0_0020 | R/W | ECC Interrupt Enable Register | 0x0000_0000 |
| 0x0_0024 | R/W1C | ECC Interrupt Status Register | 0x0000_0000 |
| 0x0_0028 | W1C | ECC Status Clear Register | 0x0000_0000 |
| 0x0_002C | R | ECC Status of Spare Region Register 0 | 0x0000_0000 |
| 0x0_0030 | R | ECC Status of Spare Region Register 1 | 0x0000_0000 |
| **NANDC Control** | | | |
| 0x0_0100 | R | Device Busy/Ready Status Register | 0x0000_00FF |
| 0x0_0104 | R/W | NANDC General Setting Register | 0x0000_6000 |
| 0x0_0108 | R/W | Memory Attribute Setting Register | 0x0000_0000 |
| 0x0_010C | R/W | Spare Region Access Mode register | 0x0000_0000 |
| 0x0_0110 | R/W | AC Timing Register 0 of NANDC 0 | 0x1F0F_1F0F |
| 0x0_0114 | R/W | AC Timing Register 1 of NANDC 0 | 0x000F_FFFF |
| 0x0_0118 | R/W | AC Timing Register 0 of NANDC 1 | 0x1F0F_1F0F |
| 0x0_011C | R/W | AC Timing Register 1 of NANDC 1 | 0x000F_FFFF |
| 0x0_0120 | R/W | AC Timing Register 0 of NANDC 2 | 0x1F0F_1F0F |
| 0x0_0124 | R/W | AC Timing Register 1 of NANDC 2 | 0x000F_FFFF |
| 0x0_0128 | R/W | AC Timing Register 0 of NANDC 3 | 0x1F0F_1F0F |
| 0x0_012C | R/W | AC Timing Register 1 of NANDC 3 | 0x000F_FFFF |
| 0x0_0130 | R/W | AC Timing Register 0 of NANDC 4 | 0x1F0F_1F0F |

| Address Offset | Type | Description | Reset Value |
|---|---|---|---|
| 0x0_0134 | R/W | AC Timing Register 1 of NANDC 4 | 0x000F_FFFF |
| 0x0_0138 | R/W | AC Timing Register 0 of NANDC 5 | 0x1F0F_1F0F |
| 0x0_013C | R/W | AC Timing Register 1 of NANDC 5 | 0x000F_FFFF |
| 0x0_0140 | R/W | AC Timing Register 0 of NANDC 6 | 0x1F0F_1F0F |
| 0x0_0144 | R/W | AC Timing Register 1 of NANDC 6 | 0x000F_FFFF |
| 0x0_0148 | R/W | AC Timing Register 0 of NANDC 7 | 0x1F0F_1F0F |
| 0x0_014C | R/W | AC Timing Register 1 of NANDC 7 | 0x000F_FFFF |
| 0x0_0150 | R/W | NANDC Interrupt Enable Register | 0x0000_0000 |
| 0x0_0154 | R/W1C | NANDC Interrupt status Register | 0x0000_0000 |
| 0x0_0158 | R | Current Access Row Address of Channel 0 | 0x0000_0000 |
| 0x0_015C | R | Current Access Row Address of Channel 1 | 0x0000_0000 |
| 0x0_0160 | R | Current Access Row Address of Channel 2 | 0x0000_0000 |
| 0x0_0164 | R | Current Access Row Address of Channel 3 | 0x0000_0000 |
| 0x0_0168 | R | Current Access Row Address of Channel 4 | 0x0000_0000 |
| 0x0_016C | R | Current Access Row Address of Channel 5 | 0x0000_0000 |
| 0x0_0170 | R | Current Access Row Address of Channel 6 | 0x0000_0000 |
| 0x0_0174 | R | Current Access Row Address of Channel 7 | 0x0000_0000 |
| 0x0_0178 | R | Read Status Register 0 | 0x0000_0000 |
| 0x0_017C | R | Read Status Register 1 | 0x0000_0000 |
| 0x0_0180 | R/W | Address Toggle Bit Location | 0x0000_0000 |
| 0x0_0184 | W | NANDC Software Reset Register | - |
| 0x0_0188 | R/W | NANDC Auto Compare Pattern Register | 0x0000_0000 |
| 0x0_0190 | R/W | AC Timing Register 2 of NANDC 0 | 0x7F7F_7F7F |
| 0x0_0194 | R/W | AC Timing Register 3 of NANDC 0 | 0xFF1F_1FFF |
| 0x0_0198 | R/W | AC Timing Register 2 of NANDC 1 | 0x7F7F_7F7F |
| 0x0_019C | R/W | AC Timing Register 3 of NANDC 1 | 0xFF1F_1FFF |
| 0x0_01A0 | R/W | AC Timing Register 2 of NANDC 2 | 0x7F7F_7F7F |
| 0x0_01A4 | R/W | AC Timing Register 3 of NANDC 2 | 0xFF1F_1FFF |
| 0x0_01A8 | R/W | AC Timing Register 2 of NANDC 3 | 0x7F7F_7F7F |
| 0x0_01AC | R/W | AC Timing Register 3 of NANDC 3 | 0xFF1F_1FFF |
| 0x0_01B0 | R/W | AC Timing Register 2 of NANDC 4 | 0x7F7F_7F7F |
| 0x0_01B4 | R/W | AC Timing Register 3 of NANDC 4 | 0xFF1F_1FFF |
| 0x0_01B8 | R/W | AC Timing Register 2 of NANDC 5 | 0x7F7F_7F7F |
| 0x0_01BC | R/W | AC Timing Register 3 of NANDC 5 | 0xFF1F_1FFF |

| Address Offset | Type | Description | Reset Value |
|---|---|---|---|
| 0x0_01C0 | R/W | AC Timing Register 2 of NANDC 6 | 0x7F7F_7F7F |
| 0x0_01C4 | R/W | AC Timing Register 3 of NANDC 6 | 0xFF1F_1FFF |
| 0x0_01C8 | R/W | AC Timing Register 2 of NANDC 7 | 0x7F7F_7F7F |
| 0x0_01CC | R/W | AC Timing Register 3 of NANDC 7 | 0xFF1F_1FFF |
| **Command Queue Control** | | | |
| 0x0_0200 | R | Command Queue Status Register | 0x0000_00FF |
| 0x0_0204 | W | Command Queue Flush Register | - |
| 0x0_0208 | R | Command Complete Counter | 0x0000_0000 |
| 0x0_020C | W | Command Complete Counter Reset Register | - |
| 0x0_0280 ~ 0x0_0294 | W | General Command Queue Access | - |
| **NANDC0 Command Queue** | | | |
| 0x0_0300 ~ 0x0_0314 | R/W | Command Queue 0 | 0x0000_0000 (All) |
| **NANDC1 Command Queue** | | | |
| 0x0_0320 ~ 0x0_0334 | R/W | Command Queue 1 | 0x0000_0000 (All) |
| **NANDC2 Command Queue** | | | |
| 0x0_0340 ~ 0x0_0354 | R/W | Command Queue 2 | 0x0000_0000 (All) |
| **NANDC3 Command Queue** | | | |
| 0x0_0360 ~ 0x0_0374 | R/W | Command Queue 3 | 0x0000_0000 (All) |
| **NANDC4 Command Queue** | | | |
| 0x0_0380 ~ 0x0_0394 | R/W | Command Queue 4 | 0x0000_0000 (All) |
| **NANDC5 Command Queue** | | | |
| 0x0_03A0 ~ 0x0_03B4 | R/W | Command Queue 5 | 0x0000_0000 (All) |
| **NANDC6 Command Queue** | | | |
| 0x0_03C0 ~ 0x0_03D4 | R/W | Command Queue 6 | 0x0000_0000 (All) |
| **NANDC7 Command Queue** | | | |
| 0x0_03E0 ~ 0x0_03F4 | R/W | Command Queue 7 | 0x0000_0000 (All) |
| **BMC Control** | | | |
| 0x0_0400 | R | Region Status Register | 0xFF00_00FF |
| 0x0_0404 | W | Region 0 User Mode Pointer Adjustment Register | - |
| 0x0_0408 | W | Region 1 User Mode Pointer Adjustment Register | - |
| 0x0_040C | W | Region 2 User Mode Pointer Adjustment Register | - |
| 0x0_0410 | W | Region 3 User Mode Pointer Adjustment Register | - |
| 0x0_0414 | W | Region 4 User Mode Pointer Adjustment Register | - |
| 0x0_0418 | W | Region 5 User Mode Pointer Adjustment Register | - |

| Address Offset | Type | Description | Reset Value |
|---|---|---|---|
| 0x0_041C | W | Region 6 User Mode Pointer Adjustment Register | - |
| 0x0_0420 | W | Region 7 User Mode Pointer Adjustment Register | - |
| 0x0_0424 | W | DMA Mode Write Data Fill/Read Data Pop Register | - |
| 0x0_0428 | W | Region Software Reset Register | - |
| 0x0_042C | R/W | Force Region Fill Read Data Register | 0x0000_0000 |
| 0x0_0430 | R | Region 0 remaining sector count of read data | 0x0000_0000 |
| 0x0_0434 | R | Region 1 remaining sector count of read data | 0x0000_0000 |
| 0x0_0438 | R | Region 2 remaining sector count of read data | 0x0000_0000 |
| 0x0_043C | R | Region 3 remaining sector count of read data | 0x0000_0000 |
| 0x0_0440 | R | Region 4 remaining sector count of read data | 0x0000_0000 |
| 0x0_0444 | R | Region 5 remaining sector count of read data | 0x0000_0000 |
| 0x0_0448 | R | Region 6 remaining sector count of read data | 0x0000_0000 |
| 0x0_044C | R | Region 7 remaining sector count of read data | 0x0000_0000 |
| **Miscellaneous** | | | |
| 0x0_0500 | R | Revision Number Register | TBD |
| 0x0_0504 | R | Feature Register | TBD |
| 0x0_0508 | R/W | AHB Slave Memory Space Range Register | 0x0280_FF01 |
| 0x0_050C | W | Global Software Reset Register | - |
| 0x0_0510 | W | AHB data slave Reset Register | - |
| 0x0_514 | R | ECC Correction Capability Register | - |
| **Programmable Flow Control** | | | |
| 0x0_0600 ~ 0x0_067C | R/W | Programmable Flow Control Register | 0xxxxx_xxxx (All) |
| **Programmable OPCODE** | | | |
| 0x0_0700 ~ 0x0_0707 | R/W | Programmable OP Code Register | 0xxxxx_xxxx |
| **Spare SRAM Access Port** | | | |
| 0x0_1000 ~ 0x0_2FFF | - | The spare data is stored in .spare SRAM and is accessed through this port | - |
| **Data SRAM Access Port** | | | |
| 0x1_0000 ~ 0x1_FFFF | - | SRAM can be accessed directly through this register. | - |

## 21.5 Register Definitions

The following sub-sections provide the detailed descriptions of the control registers

### 21.5.1 ECC Status Registers 0 and 1 (Offset = 0x0000 and 0x0004)

The ECC status register reflects the number of ECC error bits. Only the maximum error bits value is recorded. To clear this register, write '1' to the ECC status clear register (Offset = 0x0028)

**Table 21-2. ECC Status Register 0 (Offset = 0x00)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | - | - | Reserved |
| [28:24] | ecc_err_no_ch3 | R | Number of ECC error bits on Channel 3 |
| [23:21] | - | - | Reserved |
| [20:16] | ecc_err_no_ch2 | R | Number of ECC error bits on Channel 2 |
| [15:13] | - | - | Reserved |
| [12:8] | ecc_err_no_ch1 | R | Number of ECC error bits on Channel 1 |
| [7:5] | - | - | Reserved |
| [4:0] | ecc_err_no_ch0 | R | Number of ECC error bits on Channel 0 |

**Table 21-3. ECC Status Register 1 (Offset = 0x04)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | - | - | Reserved |
| [28:24] | ecc_err_no_ch7 | R | Number of ECC error bits on Channel 7 |
| [23:21] | - | - | Reserved |
| [20:16] | ecc_err_no_ch6 | R | Number of ECC error bits on Channel 6 |
| [15:13] | - | - | Reserved |
| [12:8] | ecc_err_no_ch5 | R | Number of ECC error bits on Channel 5 |
| [7:5] | - | - | Reserved |
| [4:0] | ecc_err_no_ch4 | R | Number of ECC error bits on Channel 4 |

### 21.5.2 ECC Control Register (Offset = 0x0008)

The ECC control register provides following features:

1. Threshold number of ECC error bits to generate ECC interrupt
2. ECC correction capability bits
3. ECC base size (512 bytes or 1K bytes)
4. ECC function enable/disable switch
5. ECC error data blocking function mask (Sectors containing uncorrectable data can be read from the AHB data slave port without blocking)

**Table 21-4. ECC Control Register (Offset = 0x0008)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:17] | - | - | Reserved |
| 16 | ecc_base | R/W | ECC base size<br>1: ECC parity is generated and checked based on the data size of 1K bytes.<br>0: ECC parity is generated and checked based on the data size of 512 bytes. |
| [15:8] | ecc_en*x* | R/W | ECC function enable/disable switch of channel 0 ~ 7 (*x* = 0 ~ 7)<br>1: Enable the ECC function<br>0: Disable the ECC function |
| [7:0] | ecc_err_mask*x* | R/W | ECC error data blocking mask of channel 0 ~ 7 (*x* = 0 ~ 7)<br>1: Mask the ECC error data blocking function<br>0: Keep the ECC error data blocking function<br>* The ECC related interrupts remain asserting when this bit is set. |

**Table 21-5. Threshold Number of ECC Error Bits Register 0 (Offset = 0x0010)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | - | - | Reserved |
| [28:24] | ecc_thres_bits3 | R/W | Threshold number of ECC error bits of Channel 3<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [23:21] | - | - | Reserved |
| [20:16] | ecc_thres_bits2 | R/W | Threshold number of ECC error bits of Channel 2<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [15:13] | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [12:8] | ecc_thres_bits1 | R/W | Threshold number of ECC error bits of Channel 1<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [7:5] | - | - | Reserved |
| [4:0] | ecc_thres_bits0 | R/W | Threshold number of ECC error bits of Channel 0<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |

**Table 21-6. Threshold Number of ECC Error Bits Register 1 (Offset = 0x0014)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | - | - | Reserved |
| [28:24] | ecc_thres_bits7 | R/W | Threshold number of ECC error bits of Channel 7<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [23:21] | - | - | Reserved |
| [20:16] | ecc_thres_bits6 | R/W | Threshold number of ECC error bits of Channel 6<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [15:13] | - | - | Reserved |
| [12:8] | ecc_thres_bits5 | R/W | Threshold number of ECC error bits of Channel 5<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [7:5] | - | - | Reserved |
| [4:0] | ecc_thres_bits4 | R/W | Threshold number of ECC error bits of Channel 4<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |

The permitted numbers of the ECC correction capability bits are listed in the following tables.

**Table 21-7. Permitted Number of ECC Correction Capability Bits**

| Permitted Number | ECC Parity Bytes Used by One Sector |
|---|---|
| The permitted number of ECC correction capability bits is set as below:<br>3'b000: 6 bits<br>3'b001: 10 bits<br>3'b010: 12 bits<br>3'b011: 13 bits<br>3'b100: 14 bits<br>3'b101: 16 bits<br>3'b110: 17 bits<br>3'b111: 24 bits | 11<br>18<br>21<br>23<br>25<br>28<br>30<br>42 |

Note: When programming the small-page Flash, users must set the ECC correction capability to 6 bits.

**Table 21-8. Number of ECC Correction Capability Bits Register 0 (Offset = 0x0018)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:27] | - | - | Reserved |
| [26:24] | ecc_corr_bits3 | R/W | This field sets the number of ECC correction capability bits of Channel 3.<br>Please refer to Table 4-7 for the permitted number. |
| [23:19] | - | - | Reserved |
| [18:16] | ecc_corr_bits2 | R/W | This field sets the number of ECC correction capability bits of Channel 2.<br>Please refer to Table 4-7 for the permitted bits. |
| [15:11] | - | - | Reserved |
| [10:8] | ecc_corr_bits1 | R/W | This field sets the number of ECC correction capability bits of Channel 1.<br>Please refer to Table 4-7 for the permitted bits. |
| [7:3] | - | - | Reserved |
| [2:0] | ecc_corr_bits0 | R/W | This field sets the number of ECC correction capability bits of Channel 0.<br>Please refer to Table 4-7 for the permitted bits. |

**Table 21-9. Number of ECC Correction Capability Bits Register 1 (Offset = 0x001C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:29] | ecc_corr_bits7_sp | R/W | This field sets the number of the ECC correction capability bits of Channel 7 spare region.<br>3'b000 ~ 3'b111: 1 bit ~ 8 bits |
| [28:24] | ecc_corr_bits7 | R/W | This field sets the number of the ECC correction capability bits of Channel 7.<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [23:21] | ecc_corr_bits6_sp | R/W | This field sets the number of the ECC correction capability bits of Channel 6 spare region.<br>3'b000 ~ 3'b111: 1 bit ~ 8 bits |
| [20:16] | ecc_corr_bits6 | R/W | This field sets the number of the ECC correction capability bits of Channel 6.<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [15:13] | ecc_corr_bits5_sp | R/W | This field sets the number of the ECC correction capability bits of Channel 5 spare region.<br>3'b000 ~ 3'b111: 1 bit ~ 8 bits |
| [12:8] | ecc_corr_bits5 | R/W | This field sets the number of the ECC correction capability bits of Channel 5.<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |
| [7:5] | ecc_corr_bits4_sp | R/W | This field sets the number of the ECC correction capability bits of Channel 4 spare region.<br>3'b000 ~ 3'b111: 1 bit ~ 8 bits |
| [4:0] | ecc_corr_bits4 | R/W | This field sets the number of the ECC correction capability bits of Channel 4.<br>5'b00000 ~ 5'b10111: 1 bit ~ 24 bits |

### 21.5.3 ECC Interrupt Enable Register (Offset = 0x0020)

The ECC function may results in the following two kinds of interrupts:

Number of ECC error bits hit the threshold set in the ECC control register

Failure of ECC correction

**Table 21-10. ECC Interrupt Enable Register (Offset = 0x0020)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:4] | - | - | Reserved |
| 3 | ecc_err_hit_thres_en_sp | R/W | ECC error bits hit the threshold number in the spare region interrupt enable.<br>1: Enable<br>0: Disable |

| Bit | Name | Type | Description |
|---|---|---|---|
| 2 | ecc_corr_fail_en_sp | R/W | ECC correction fail of the spare region interrupt enable.<br>1: Enable<br>0: Disable |
| 1 | ecc_err_hit_thres_en | R/W | ECC error bits hit the threshold number interrupt enable.<br>1: Enable<br>0: Disable |
| 0 | ecc_corr_fail_en | R/W | ECC correction fail interrupt enable<br>1: Enable<br>0: Disable |

### 21.5.4 ECC Interrupt Status Register (Offset = 0x0024)

The ECC interrupt status register indicates the failure of ECC correction fail or the error bits hitting the threshold of each channel. Write '1' to the specific bit to clear the status.

**Table 21-11. ECC Interrupt Enable Register (Offset = 0x0024)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | ecc_err_hit_thres_ch7_sp | R/W1C | ECC error bits hit the threshold of Channel 7 spare region. |
| 30 | ecc_err_hit_thres_ch6_sp | R/W1C | ECC error bits hit the threshold of Channel 6 spare region. |
| 29 | ecc_err_hit_thres_ch5_sp | R/W1C | ECC error bits hit the threshold of Channel 5 spare region. |
| 28 | ecc_err_hit_thres_ch4_sp | R/W1C | ECC error bits hit the threshold of Channel 4 spare region. |
| 27 | ecc_err_hit_thres_ch3_sp | R/W1C | ECC error bits hit the threshold of Channel 3 spare region. |
| 26 | ecc_err_hit_thres_ch2_sp | R/W1C | ECC error bits hit the threshold of Channel 2 spare region. |
| 25 | ecc_err_hit_thres_ch1_sp | R/W1C | ECC error bits hit the threshold of Channel 1 spare region. |
| 24 | ecc_err_hit_thres_ch0_sp | R/W1C | ECC error bits hit the threshold of Channel 0 spare region. |
| 23 | ecc_err_fail_ch7_sp | R/W1C | ECC correction fail of the Channel 7 spare region |
| 22 | ecc_err_fail_ch6_sp | R/W1C | ECC correction fail of the Channel 6 spare region |
| 21 | ecc_err_fail_ch5_sp | R/W1C | ECC correction fail of the Channel 5 spare region |
| 20 | ecc_err_fail_ch4_sp | R/W1C | ECC correction fail of the Channel 4 spare region |
| 19 | ecc_err_fail_ch3_sp | R/W1C | ECC correction fail of the Channel 3 spare region |
| 18 | ecc_err_fail_ch2_sp | R/W1C | ECC correction fail of the Channel 2 spare region |
| 17 | ecc_err_fail_ch1_sp | R/W1C | ECC correction fail of the Channel 1 spare region |
| 16 | ecc_err_fail_ch0_sp | R/W1C | ECC correction fail of the Channel 0 spare region |
| 15 | ecc_err_hit_thres_ch7 | R/W1C | ECC error bits hit the threshold of Channel 7 |

| Bit | Name | Type | Description |
|---|---|---|---|
| 14 | ecc_err_hit_thres_ch6 | R/W1C | ECC error bits hit the threshold of Channel 6. |
| 13 | ecc_err_hit_thres_ch5 | R/W1C | ECC error bits hit the threshold of Channel 5. |
| 12 | ecc_err_hit_thres_ch4 | R/W1C | ECC error bits hit the threshold of Channel 4. |
| 11 | ecc_err_hit_thres_ch3 | R/W1C | ECC error bits hit the threshold of Channel 3. |
| 10 | ecc_err_hit_thres_ch2 | R/W1C | ECC error bits hit the threshold of Channel 2. |
| 9 | ecc_err_hit_thres_ch1 | R/W1C | ECC error bits hit the threshold of Channel 1. |
| 8 | ecc_err_hit_thres_ch0 | R/W1C | ECC error bits hit the threshold of Channel 0. |
| 7 | ecc_err_fail_ch7 | R/W1C | ECC correction fail of Channel 7 |
| 6 | ecc_err_fail_ch6 | R/W1C | ECC correction fail of Channel 6 |
| 5 | ecc_err_fail_ch5 | R/W1C | ECC correction fail of Channel 5 |
| 4 | ecc_err_fail_ch4 | R/W1C | ECC correction fail of Channel 4 |
| 3 | ecc_err_fail_ch3 | R/W1C | ECC correction fail of Channel 3 |
| 2 | ecc_err_fail_ch2 | R/W1C | ECC correction fail of Channel 2 |
| 1 | ecc_err_fail_ch1 | R/W1C | ECC correction fail of Channel 1 |
| 0 | ecc_err_fail_ch0 | R/W1C | ECC correction fail of Channel 0 |

### 21.5.5 ECC Status Clear Register (Offset = 0x0028)

**Table 21-12. ECC Status Clear Register (Offset = 0x0024)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [15:8] | ecc_err_cnt_sp_clr_chx | W | ECC error bits clear region of Channel x (x = 0 ~ 7) spare region<br>1: Clear error bits recorded in the ECC status register of spare region<br>0: No effect |
| [7:0] | ecc_err_cnt_clr_chx | W | ECC error bits clear register of CH*x* (*x* = 0 ~ 7)<br>1: Clear error bits recorded in the ECC status register<br>0: No effect |

### 21.5.6 ECC Status of Spare Region Registers 0 and 1 (Offset = 0x002C and 0x0030)

The ECC status of the spare region register reflects the number of the ECC error bits happened in the spare region. Only the values of the maximum error bits are recorded. To clear this register, write '1' to the ECC status clear register bit 8 to bit 15 (Offset = 0x0028)

**Table 21-13. ECC Status of Spare Region Register 0 (Offset = 0x2C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:28] | - | - | Reserved |
| [27:24] | ecc_err_no_ch3_sp | R | Number of the ECC error bits on Channel 3 spare region |
| [23:20] | - | - | Reserved |
| [19:16] | ecc_err_no_ch2_sp | R | Number of the ECC error bits on Channel 2 spare region |
| [15:12] | - | - | Reserved |
| [11:8] | ecc_err_no_ch1_sp | R | Number of the ECC error bits on Channel 1 spare region |
| [7:4] | - | - | Reserved |
| [3:0] | ecc_err_no_ch0_sp | R | Number of the ECC error bits on Channel 0 spare region |

**Table 21-14. ECC Status of Spare Region Register 1 (Offset = 0x30)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:28] | - | - | Reserved |
| [27:24] | ecc_err_no_ch7_sp | R | Number of the ECC error bits on Channel 7 spare region |
| [23:20] | - | - | Reserved |
| [19:16] | ecc_err_no_ch6_sp | R | Number of the ECC error bits on Channel 6 spare region |
| [15:12] | - | - | Reserved |
| [11:8] | ecc_err_no_ch5_sp | R | Number of the ECC error bits on Channel 5 spare region |
| [7:4] | - | - | Reserved |
| [3:0] | ecc_err_no_ch4_sp | R | Number of the ECC error bits on Channel 4 spare region |

### 21.5.7 Device Busy/Ready Status Register (Offset = 0x0100)

The Flash device busy/ready pins can be monitored by reading this register.

**Table 21-15. Device Busy/Ready Status Register (Offset = 0x0100)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | dev_busy_rdyx | R | Flash device busy/ready (busy_n) pins status of channel *x* (*x* = 0 ~ 7)<br>1: Flash is ready.<br>0: Flash is busy. |

### 21.5.8 NANDC General Setting Register (Offset = 0x0104)

This register provides general settings of the NANDC module.

**Table 21-16. NANDC General Setting Register (Offset = 0x0104)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:26] | - | - | Reserved |
| [25:24] | ce_num | R/W | Number of CE connected with Flash in a channel<br>00: 1 CE<br>01: 2 CE<br>10: 4 CE<br>11: 8 CE<br>Notes:<br>1. The number of Flashes used in a channel must be power of 2.<br>2. If two CEs are at the CE0 and CE2 positions, ce_num must be set to 4 CE. |
| [23:16] | bypass_crc_chkx | R/W | Bypass CRC check of the spare data of Channel *x* (*x* = 0 ~ 7)<br>1: Bypass the CRC check<br>0: Perform the CRC check |
| 15 | - | - | Reserved |
| [14:12] | busy_rdy_loc | R/W | Device busy/ready status bit location on the Flash data bus<br>3'b000 ~ 3'b111: Bit 0 ~ bit 7 |
| 11 | - | - | Reserved |
| [10:8] | cmd_sts_loc | R/W | Command pass/fail status bit location on the Flash data bus<br>3'b000 ~ 3'b111: Bit 0 ~ bit 7 |
| [7:5] | - | - | Reserved |
| 4 | report_addr_en | R/W | Enable the row address update of each channel to the channel row address register<br>1: Enable<br>0: Disable |
| 3 | - | - | Reserved |
| 2 | wr_prot_en | R/W | Flash write protect pin control<br>1: Enable the write protect<br>0: Disable the write protect |

| Bit | Name | Type | Description |
|---|---|---|---|
| 1 | data_inv_en | R/W | Data inverse enable<br>The inverted targets include ECC parity and data.<br>1: Enable<br>0: Disable<br>This inverse function is useful when reading data from an erased page from Flash. If the inverse function is not enabled, all the read data bits will be '1' and the ECC parity bits will also be '1', which will induce an ECC parity error. If the inverse function is enabled, all data will be '0' and the ECC parity bits will be '0'. All '0' data and parity can pass the ECC parity check without errors. |
| 0 | scrambler_en | R/W | Data scrambler<br>The scrambling operation only affects data.<br>1: Enable<br>0: Disable<br>The scrambler adopts a formula to inverse data in random pseudo. |

### 21.5.9 Memory Attribute Setting Register (Offset = 0x0108)

**Table 21-17. Memory Attribute Setting Register (Offset = 0x0108)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | - | - | Reserved |
| [30:26] | dv_size | R/W | Device size (Number of pages associated with one CE)<br>Number of pages associated with one CE = 2 ^ dv_size<br>It is affected when auto_ce_sel is set '1'. |
| [25:16] | spare_size | R/W | Spare area size<br>This value represents the total number of bytes in a page minus the number of bytes of raw data. This value is only used in the Blanking Check fixed flow.<br>(For example: If a Flash contains 4314 bytes per page, it will contain a spare area size of 4314 – 4096 = 218 bytes.) |
| 15 | - | - | Reserved |
| [14:13] | row_cyc | R/W | Flash row address cycles<br>2'b00: One cycle<br>2'b01: Two cycles<br>2'b10: Three cycles<br>2'b11: Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| 12 | col_cyc | R/W | Flash column address cycles<br>1'b0: One cycle<br>1'b1: Two cycles |
| [11:2] | block_size | R/W | Block size of Flash (Number of pages contained in one block)<br>10'h0 ~ 10'h3FF: 1 page ~ 1024 pages |
| [1:0] | page_size | R/W | Page size of Flash<br>2'b00: 512 byte (Small page)<br>2'b01: 2K bytes<br>2'b10: 4K bytes<br>2'b11: 8K bytes<br>Note: The small page fixed flow is only used for the small page setting. |

### 21.5.10 Spare Region Access Mode Register (Offset = 0x010C)

**Table 21-18. Spare Region Access Mode Register (Offset = 0x010C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | spare_protect_en | R/W | Spare data protect enable<br>1: Enable protection by CRC or ECC<br>0: Disable protection |
| [7:0] | spare_access_mode | R/W | Spare data access mode<br>1: Page mode (ECC base)<br>0: Sector mode (CRC base) → Read/Write flow CRC ON/OFF must be consistent.<br>Please refer to Table 4-39 for details.<br>Notes: spare_access_mode = 0 and byte_mode = 0 is allowed.<br>spare_access_mode = 0 and byte mode = 1 is allowed.<br>spare_access mode = 1 and byte mode = 0 is allowed.<br>spare access mode = 1 and byte mode = 1 is not allowed. |

### 21.5.11 Flash AC Timing Register 0 of Channel *x* (Offset = 0x0110 + *x**0x8, *x* = 0 ~ 7)

The AC timing registers 0 and 1 decide the clock cycles that will be issued by a specific signal pulse. The reference clock is nand_clk.

**Table 21-19. Flash AC Timing Register 0 (Offset = 0x0110 + *x**0x8, *x* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:28] | - | - | Reserved |
| [27:24] | $t_{WH\_nx}$ | R/W | WE_n high pulse width |
| [23:21] | - | - | Reserved |
| [20:16] | $t_{WP\_nx}$ | R/W | WE_n low pulse width |
| [15:12] | - | - | Reserved |
| [11:8] | $t_{REH\_nx}$ | R/W | RE_n high pulse width |
| [7:5] | - | - | Reserved |
| [4:0] | $t_{RES\_nx}$ | R/W | Maximum pulse width of $t_{REA}$/$t_{RSTO}$/$t_{READ}$<br>($t_{RP}$ will be $t_{RES\_nx}$ in the EDO mode.) |

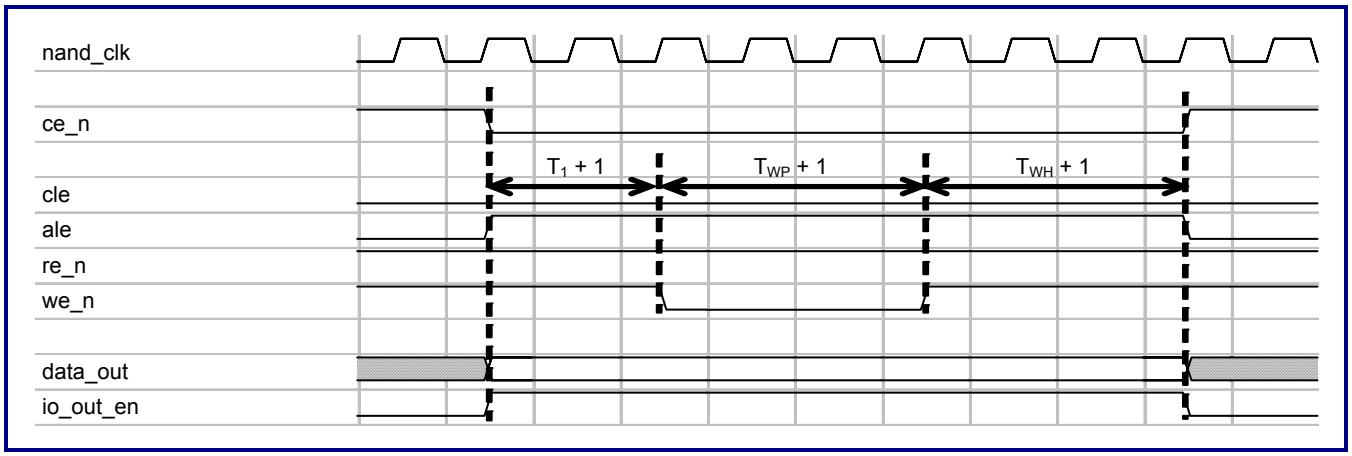



**Figure 21-2. Timing Diagram of Address State**

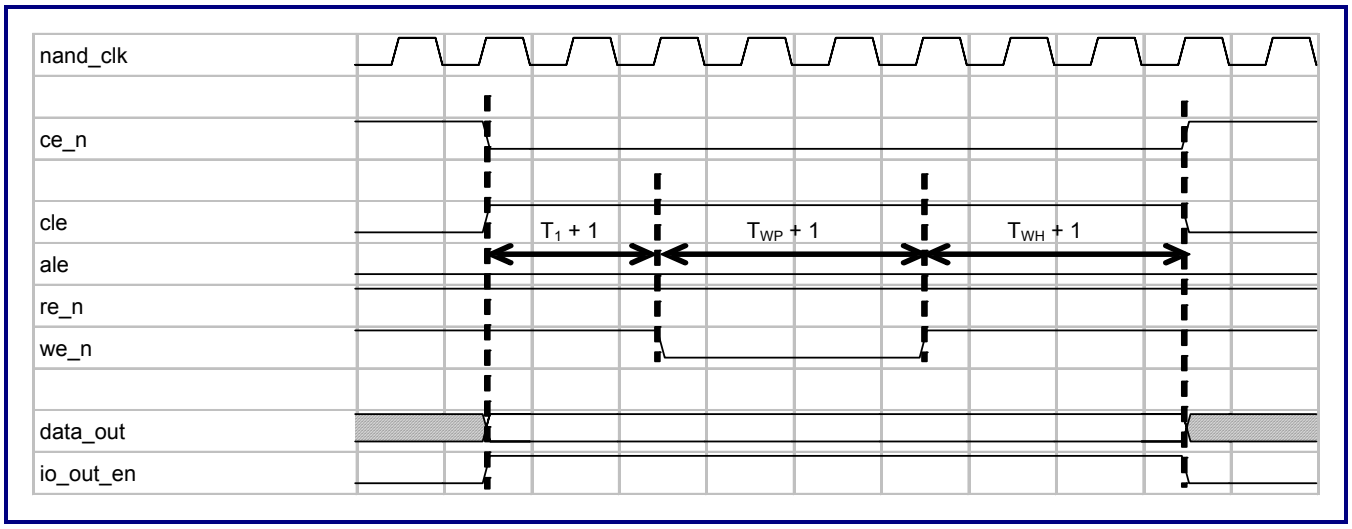



**Figure 21-3. Timing Diagram of Command State**

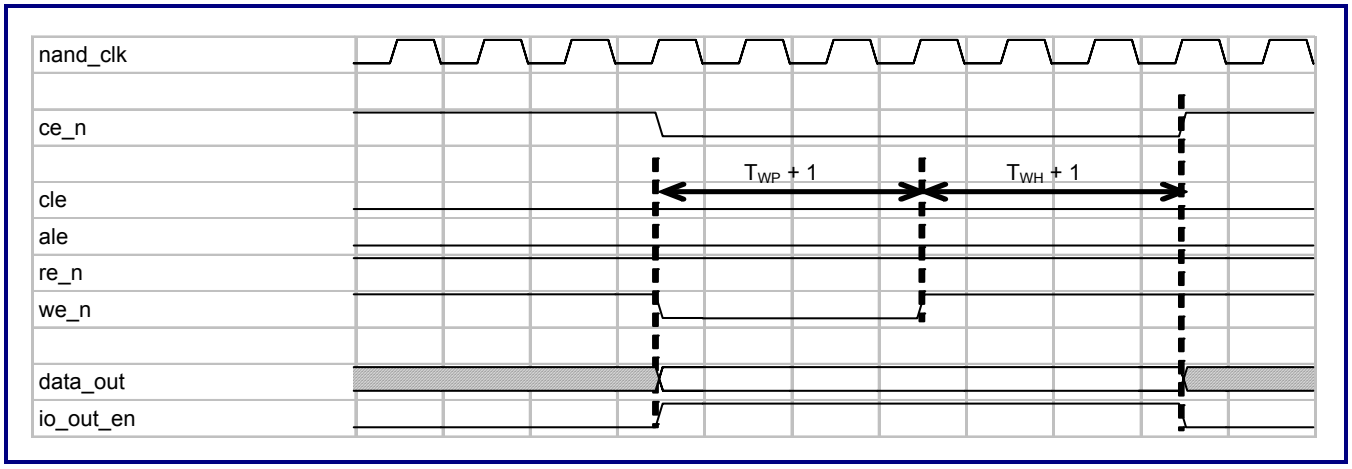



**Figure 21-4. Timing Diagram of Write Data State**

## 21.5.12 Flash AC Timing Register 1 of Channel *x* (Offset = 0x0114 + *x**0x8, *x* = 0 ~ 7)

**Table 21-20. Flash AC Timing Register 1 (Offset = 0x0114 + *x**0x8, *x* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:22] | - | - | Reserved |
| [21:16] | $t_{RLAT\_nx}$ | R/W | Internal read data latch ($t_{RLAT}$ < ($t_{RES}$ + $t_{REH}$ + 2)) |
| 15 | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [14:8] | $t_{BSY\_nx}$ | R/W | Maximum value of $t_{WB}/t_{RB}$<br>The minimum possible value of $t_{BSY}$ is 1. |
| 7 | - | - | Reserved |
| [6:0] | t1_nx | R/W | The maximum value of $t_{CS}/t_{CLS}/t_{ALS}$ minus $t_{WP.}$ |

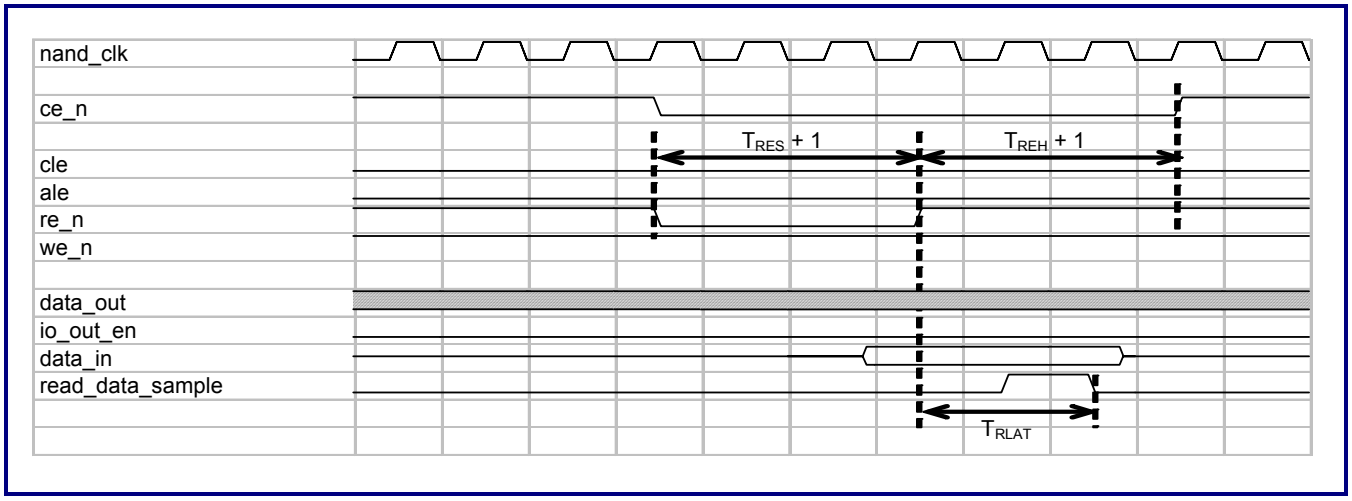



**Figure 21-5. Timing Diagram of Read Data State**

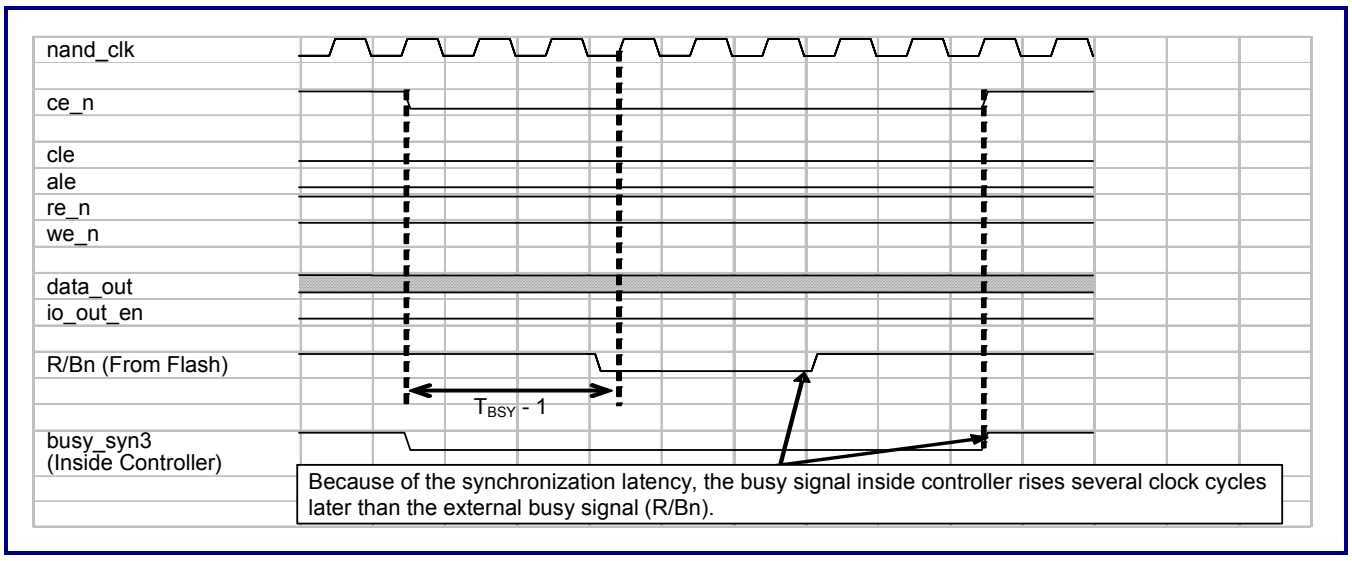



**Figure 21-6. Timing Diagram of Busy State**

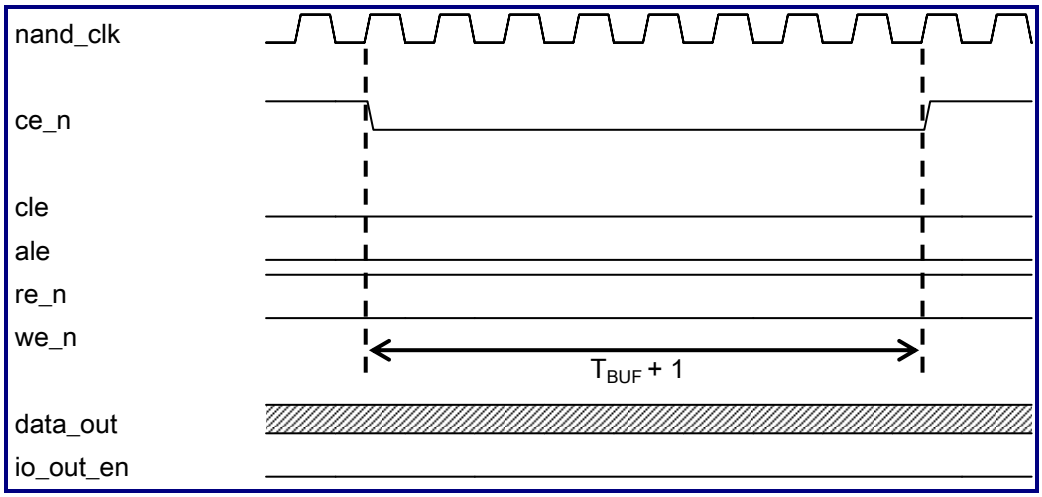



**Figure 21-7. Timing Diagram of Buffer State**

## 21.5.13 Flash AC Timing Register 2 of Channel *x* (Offset = 0x0190 + *x* * 0x8, *x* = 0 ~ 7)

**Table 21-21. Flash AC Timing Register 2 (Offset = 0x0190 + *x**0x8, *x* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | - | - | Reserved |
| [30:24] | $t_{BUF4\_nx}$ | R/W | Maximum value of $t_{WHR}$<br>Minimum value is 1. |
| 23 | - | - | Reserved |
| [22:16] | $t_{BUF3\_nx}$ | R/W | Maximum value of $t_{RHW}$/$t_{RHZ}$/$t_{DQSHZ}$<br>Minimum value is 2 times ns_clk/sys_clk (sys_clk is slow). |
| 15 | - | - | Reserved |
| [14:8] | $t_{BUF2\_nx}$ | R/W | Maximum value of $t_{AR}$/$t_{RR}$/$t_{CLR}$<br>Minimum value is 1. |
| 7 | - | - | Reserved |
| [6:0] | tBUF1_nx | R/W | Maximum value of tADL/tCCS<br>Minimum value is 1 |

### a. Timing Diagram of Micron ONFi

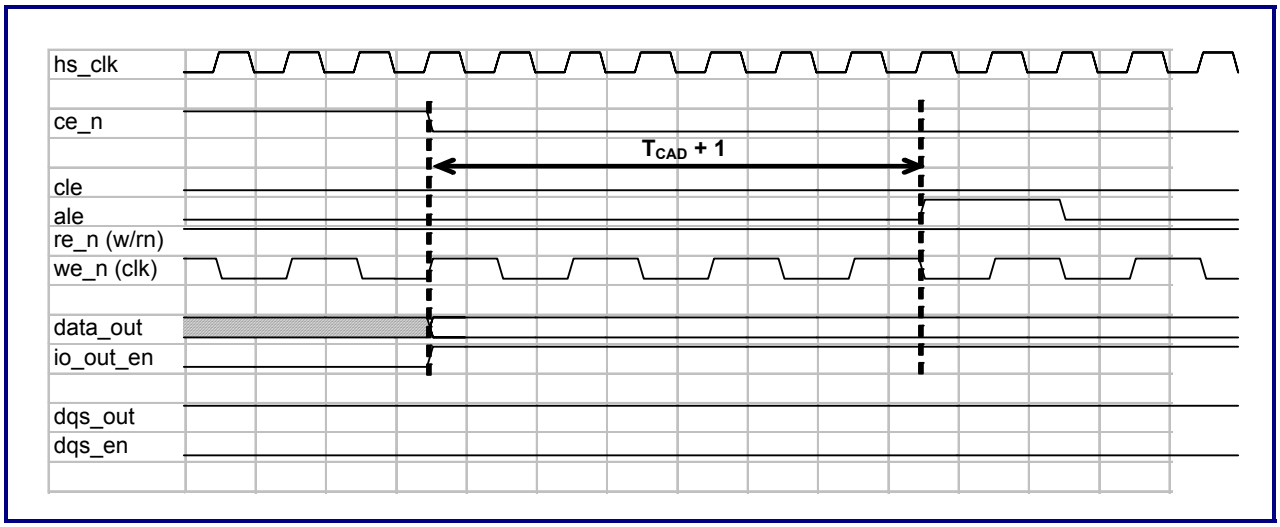



**Figure 21-8. Timing Diagram of Address State**

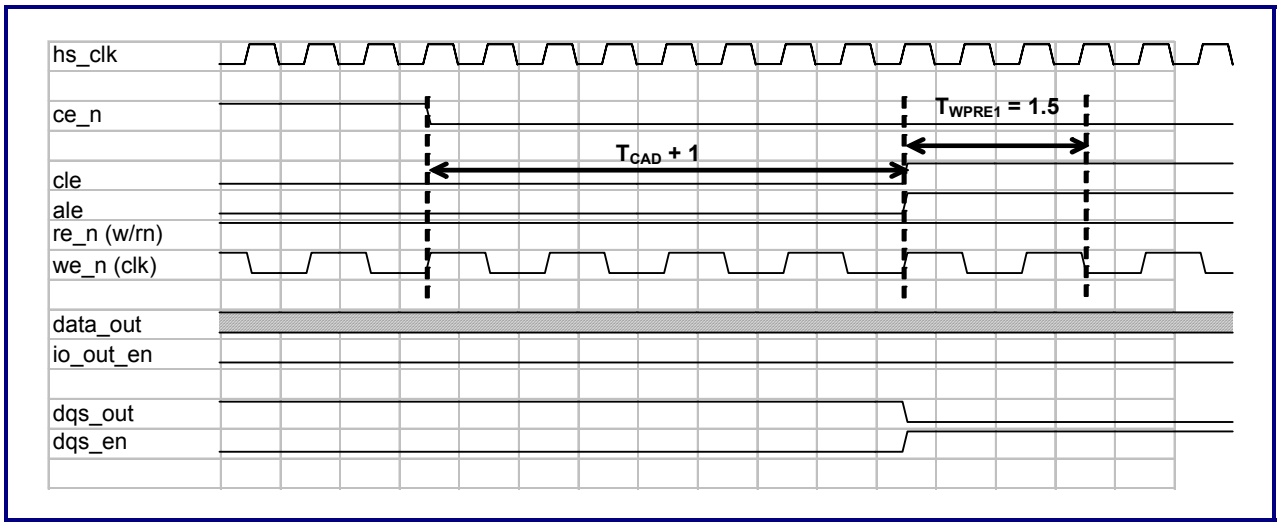



**Figure 21-9. Timing Diagram of Pre-amble Write State**

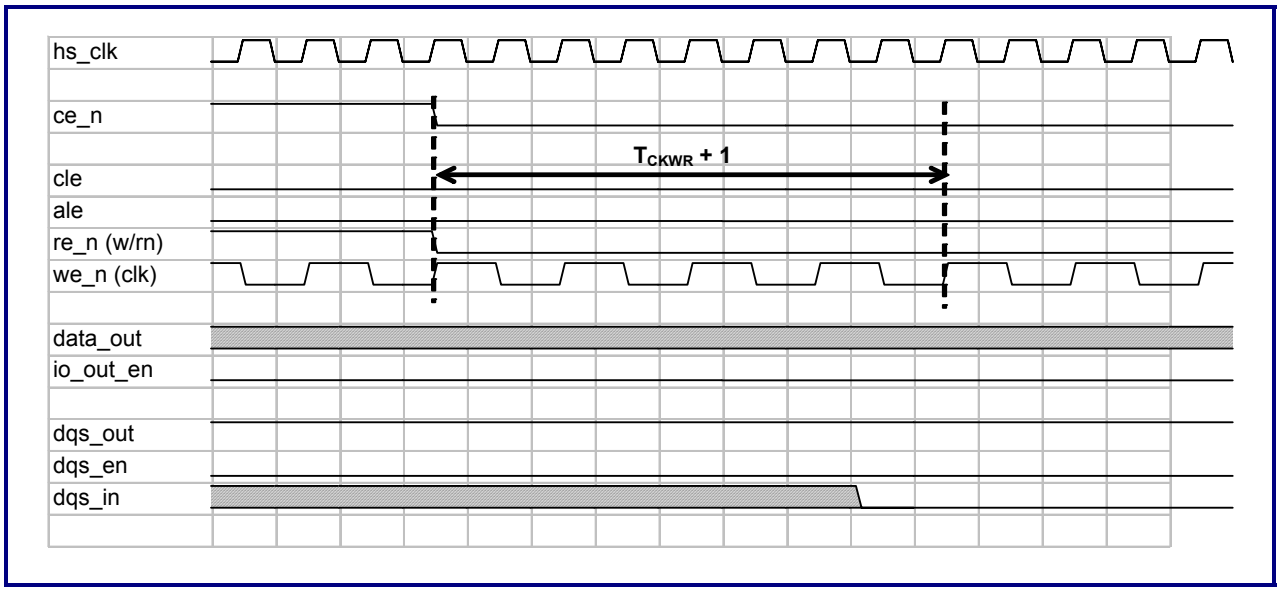



**Figure 21-10. Timing Diagram of Pre-amble Read State**

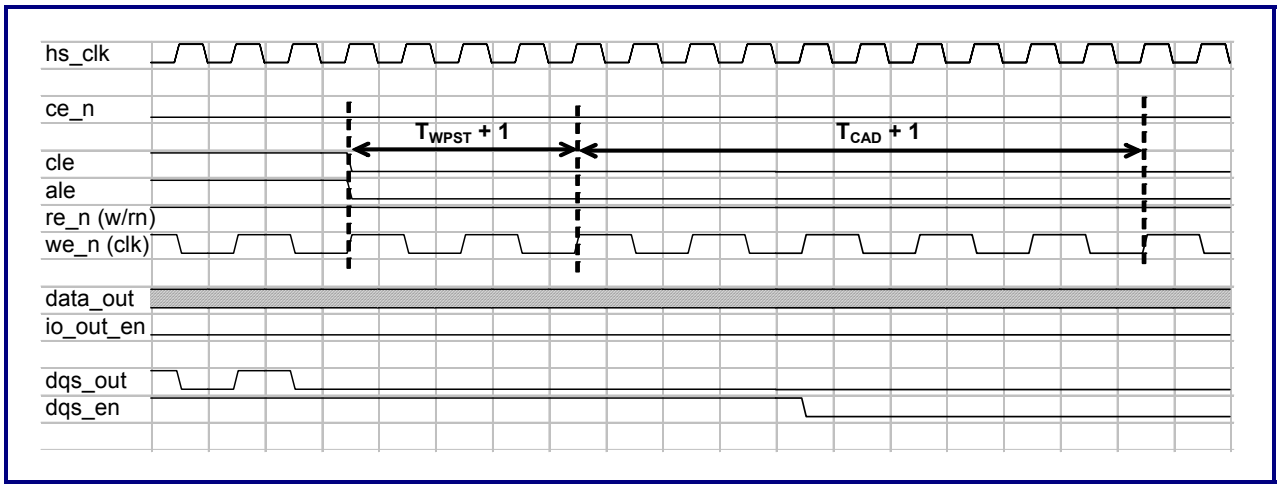



**Figure 21-11. Timing Diagram of Post-amble Write State**

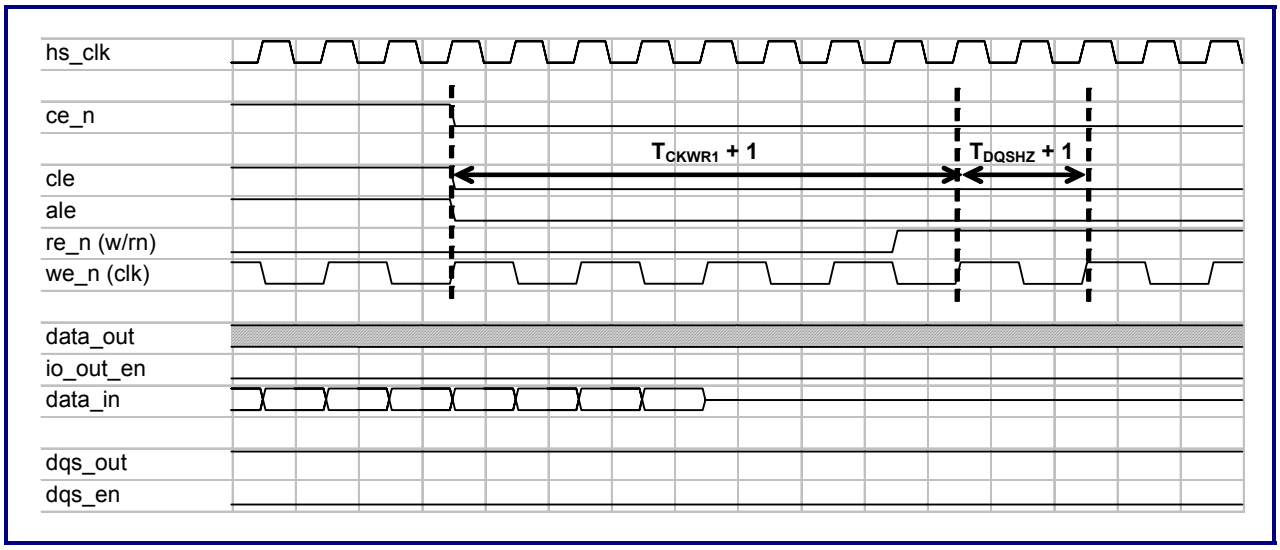



**Figure 21-12. Timing Diagram of Post-amble Read State**

### b. Timing Diagram of Samsung Toggle

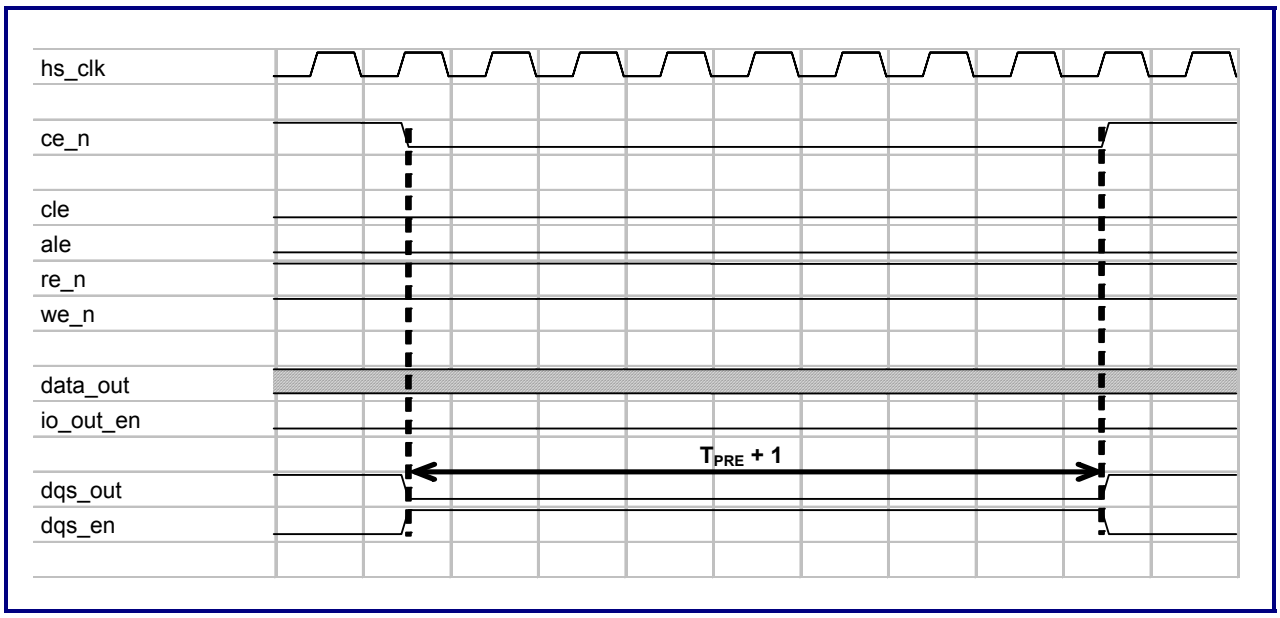



**Figure 21-13. Timing Diagram of Pre-amble Write State**

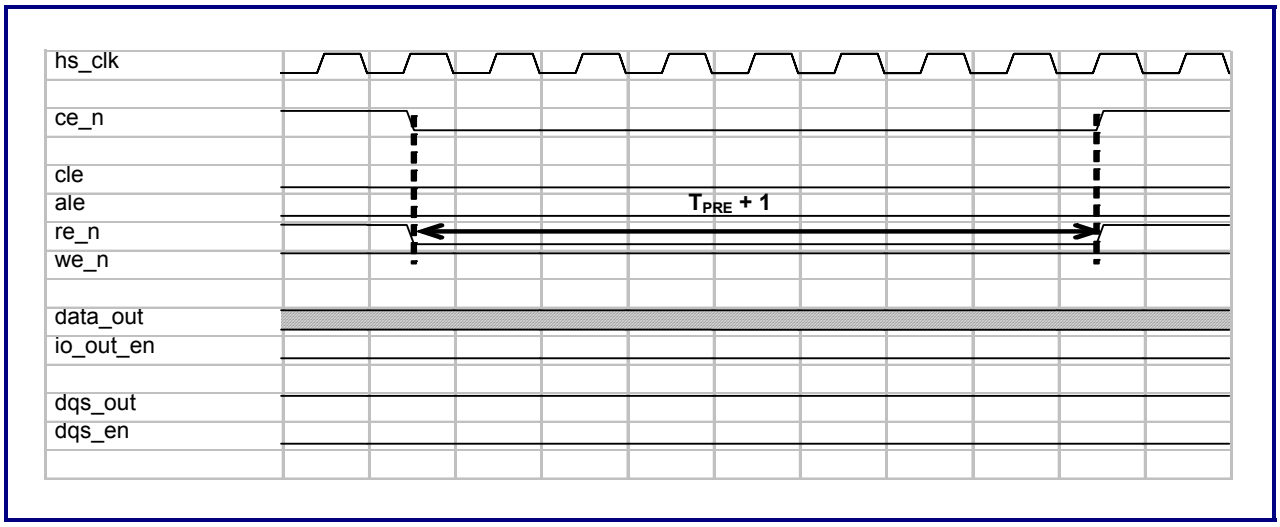



**Figure 21-14. Timing Diagram of Pre-amble Read State**

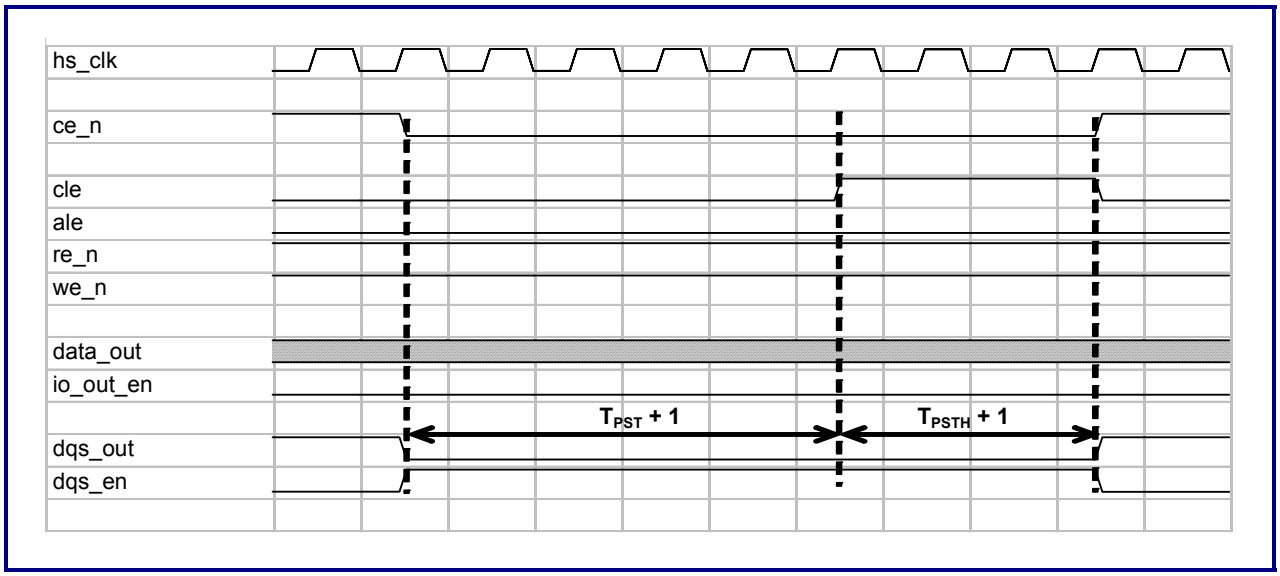



**Figure 21-15. Timing Diagram of Post-amble Write State**

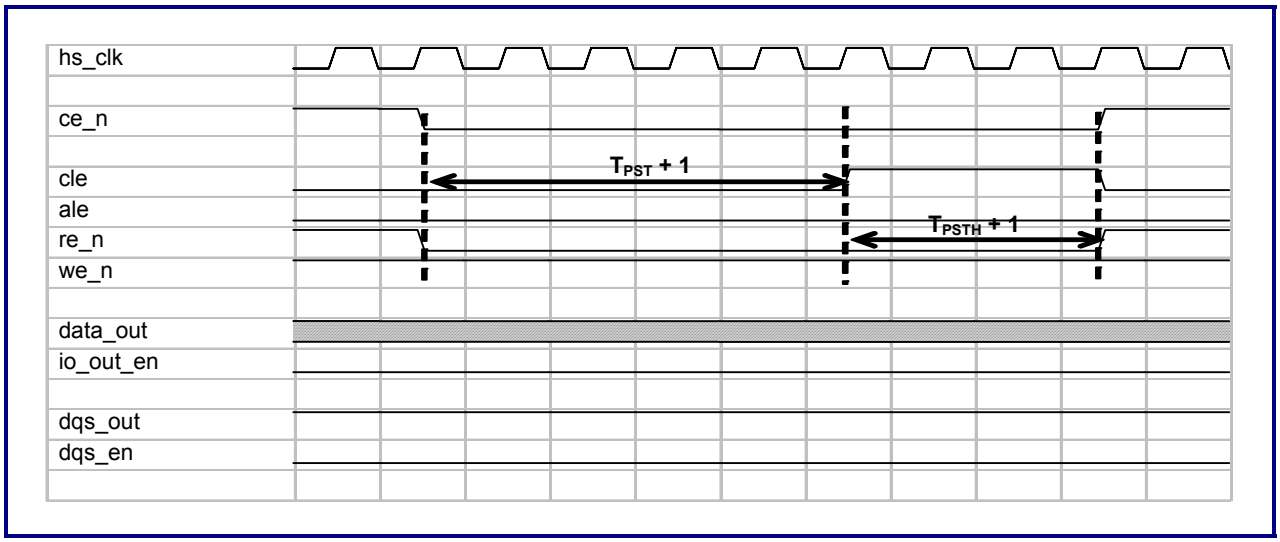



**Figure 21-16. Timing Diagram of Post-amble Read State**

**Summary of AC Timing Usage**

| Item | Asynchronous IF | Toggle IF | Synchronous IF |
|---|---|---|---|
| $t_{WH}$ | $t_{WH}$ | $t_{WH}$ | — |
| $t_{WP}$ | $t_{WP}$ | $t_{WP}$ | — |
| $t_{REH}$ | $t_{REH}$ | — | — |
| $t_{RES}$ | $t_{REA}$,$t_{RP}$ | — | — |
| $t_{BSY}$ | $t_{WB}$,$t_{RB}$ | $t_{WB}$ | $t_{WB}$ |
| $t_{BUF1}$<br>$t_{BUF2}$<br>$t_{BUF3}$<br>$t_{BUF4}$ | $t_{WHR}$,$t_{RHW}$,<br>$t_{RR}$,$t_{AR}$,<br>$t_{ADL}$,$t_{RHZ}$,$t_{CCS}$ | $t_{WHR}$,$t_{WHR2}$,$t_{RHW}$,<br>$t_{RR}$,$t_{AR}$,<br>$t_{ADL}$,$t_{RHZ}$, | $t_{WHR}$,$t_{RHW}$,<br>$t_{RR}$,$t_{ADL}$ |
| $t_{RLAT}$ | internal latch timing | — | — |
| $t_{I}$ | max($t_{CS}$,$t_{CLS}$,$t_{ALS}$) - $t_{WP}$ | max($t_{CALS2}$ - $t_{WP}$,$t_{CWAW}$) | — |
| $t_{PRE}$ | — | $t_{WPRE}$,$t_{RPRE}$ | $t_{WPRE}$ |
| $t_{PST}$ | — | $t_{WPST}$,$t_{RPST}$ | $t_{WPST}$ |
| $t_{PSTH}$ | — | $t_{WPSTH}$,$t_{RPSTH}$ | $t_{DQSHZ}$ |
| $t_{CAD}$ | — | max($t_{RP}$,$t_{REH}$) | $t_{CAD}$ |
| $t_{CS}$ | — | — | $t_{CS}$ |
| $t_{CKWR}$ | — | max($t_{DSL}$,$t_{DSH}$,$t_{DQSL}$,$t_{DQSH}$) | $t_{CKWR}$,$t_{WRCK}$ |
| | | | |
| NOTE : 1. ONFi 2.0 protocol includes Asynchronous IF and Synchronous IF | | | |
| 2. Legacy protocol means that Asynchronous IF | | | |

**Figure 21-17. Summary of AC Timing Usage**

### 21.5.14 NANDC Interrupt Enable Register (Offset = 0x0150)

The interrupt of FTNANDC023 is asserted when the interrupt enable is set and the corresponded event happens. The event status can be checked by reading the NANDC Interrupt Status Register (Offset = 0x0154).

**Table 21-22. NANDC Interrupt Enable Register (Offset = 0x0150)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | crc_fail_int_en | R/W | CRC check fail interrupt enable<br>1: Enable<br>0: Disable<br>(Bit 15 ~ Bit 8: Channel 7 ~ Channel 0) |
| [7:0] | sts_fail_int_en | R/W | Flash status check fail interrupt enable<br>1: Enable<br>0: Disable<br>(Bit 7 ~ Bit 0: Channel 7 ~ Channel 0) |

### 21.5.15 NANDC Interrupt Status Register (Offset = 0x0154)

There are four kinds of NANDC events that can assert the FTNANDC023 interrupt.

- Auto pattern compare failed
- NANDC command completed
- CRC check failed
- Status check failed

**Table 21-23. NANDC Interrupt Status Register (Offset = 0x0154)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | crc_fail | R/W1C | CRC check fail<br>(Bit 15 ~ Bit 8: Channel 7 ~ Channel 0) |
| [7:0] | sts_fail | R/W1C | Status check fail<br>(Bit 7 ~ Bit 0: Channel 7 ~ Channel 0) |

### 21.5.16 NANDC Interrupt Status Register (Offset = 0x0154)

The FTNANDC023 interrupt can be asserted by using one of the following four NANDC events.

- Auto pattern compare failed
- NANDC command completed
- CRC check failed
- Status check failed

**Table 21-24. NANDC Interrupt Status Register (Offset = 0x0154)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | auto_cmp_pat_fail Note1 | R/W1C | Auto pattern compare fail<br>(Bit 31 ~ Bit 24: Channel 7 ~ Channel 0) |
| [23:16] | nandc_cmd_cmplt Note2 | R/W1C | NANDC command complete<br>(Bit 23 ~ Bit 16: Channel 7 ~ Channel 0) |
| [15:8] | crc_fail | R/W1C | CRC check fail<br>(Bit 15 ~ Bit 8: Channel 7 ~ Channel 0) |
| [7:0] | sts_fail | R/W1C | Status check fail<br>(Bit 7 ~ Bit 0: Channel 7 ~ Channel 0) |

Notes:

1. The failure of the auto pattern compare always triggers the global interrupt.
2. The NANDC command complete interrupt enable is set in the command queue.
3. When auto_cmp_pat_fail occurs, nandc_cmd_cmplt will not be asserted.

### 21.5.17 Current Access Row Address Register (Offset = 0x0158 ~ 0x0174)

The current access row address represents the row address that is issued by a NANDC. These registers are updated once a NANDC issues a new row address and report_addr_en (Offset = 0x0104) is set.

**Table 21-25. Current Access Row Address Register (Offset = 0x0158 ~ 0x0174)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | row_addr_ch*x** | R | Current access row address of channel *x*[Note] |

Note: *x* = 0 ~ 7 (Corresponding to offsets 0x0158 ~ 0x0174)

### 21.5.18 Read Status Registers, 0 and 1 (Offset = 0x0178 ~ 0x017C)

NANDC stores the status returned from Flash in the register.

**Table 21-26. Read Status Register 0 (Offset = 0x0178)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | read_sts_ch3 | R | Flash status of Channel 3 |
| [23:16] | read_sts_ch2 | R | Flash status of Channel 2 |
| [15:8] | read_sts_ch1 | R | Flash status of Channel 1 |
| [7:0] | read_sts_ch0 | R | Flash status of Channel 0 |

**Table 21-27. Read Status Register 1 (Offset = 0x017C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | read_sts_ch7 | R | Flash status of Channel 7 |
| [23:16] | read_sts_ch6 | R | Flash status of Channel 6 |
| [15:8] | read_sts_ch5 | R | Flash status of Channel 5 |
| [7:0] | read_sts_ch4 | R | Flash status of Channel 4 |

### 21.5.19 Address Toggle Bit Location Register (Offset = 0x0180)

The row address may need to be toggled for one-bit in order to perform the Flash two-plane access. This register can be set to determine which bit is toggled during the two-plane access.

**Table 21-28. Address Toggle Bit Location Register (Offset = 0x0180)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | - | - | Reserved |
| [23:0] | toggle_bit_loc | R/W | Toggle bit location<br>0 ~ 23: Bit 0 ~ Bit 23 of the row addresses<br>Others: Reserved |

### 21.5.20 NANDC Software Reset Register (Offset = 0x0184)

Each NANDC can be reset by setting this register. The NANDC will return to the idle state when performing reset. The corresponding command queue will not pop and will be re-executed immediately after reset. No register value will be reset during issuing the software reset to NANDC.

**Table 21-29. NANDC Software Reset Register (Offset = 0x0184)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | nandc_sw_rst | R/W | NANDC software reset register<br>This reset is cleared when the corresponding NANDC finishes the reset operation. This field must be checked to be '0' before issuing the new NAND commands.<br>Write 1: Reset<br>Write 0: No effect<br>(Bit 7 ~ Bit 0: Channel 7 ~ Channel 0) |

## 21.5.21 NANDC Auto-compare Pattern Register (Offset = 0x0188)

NANDC can perform a write command with specific patterns and read commands to be compared with the specific pattern recorded in this register. Please note that the ECC enable must not be set during performing the auto-compare pattern (Blanking check) operation.

**Table 21-30. NANDC Auto-compare Pattern Register (Offset = 0x0188)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:28] | - | - | Reserved |
| [27:24] | spare_cmp_thres | R/W | Compare threshold of error bits in a spare region*<br>Auto compare fail interrupt asserts when the error bits in the spare region is equal to or exceeds the threshold.<br>4'b0000 ~ 4'b1111: 1 bit ~ 16 bits<br>*A spare region is a space in a page that is not occupied by data or ECC parity. |
| [23:21] | - | - | Reserved |
| [20:16] | cmp_thres | R/W | Compare threshold of error bits in a normal data region<br>Auto compare fail interrupt asserts when the error bits in a sector is equal to or exceeds the threshold.<br>5'b00000 ~ 5'b11000: 1 bit ~ 25 bits<br>Others: Reserved |
| [15:0] | auto_cmp_pat | R/W | Auto compare pattern |

### 21.5.22 Command Queue Status Register (Offset = 0x0200)

Each NANDC has its own command queue. The command queue status includes:

- Command queue full
- Command queue empty

Before pushing a command into queue, check the queue to make sure it is not full.

**Table 21-31. Command Queue Status Register (Offset = 0x0200)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | cmdq_full | R | Command queue full status<br>1: Full<br>0: Not full<br>(Bit 15 ~ Bit 8: Channel 7 ~ Channel 0) |
| [7:0] | cmdq_empty | R | Command queue empty status<br>1: Empty<br>0: Not empty<br>(Bit 7 ~ Bit 0: Channel 7 ~ Channel 0) |

### 21.5.23 Command Queue Flush Register (Offset = 0x0204)

The command queue can be flushed by writing '1' to the specific flush register. The corresponded NANDC is also reset during setting the command queue flush.

**Table 21-32. Command Queue Flush Register (Offset = 0x0204)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | cmdq_flush | W | Command queue flush<br>1: Write 1 to flush<br>0: Write 0 has no effect<br>(Bit 7 ~ Bit 0: Channel 7 ~ Channel 0) |

### 21.5.24 Command Complete Counter (Offset = 0x0208)

When one command completes, the counter will be increased by 1. By checking this register, users can get the number of the complete commands. Please note that cmd_cmplt_cnt contains a maximum of seven cmd_cmplt, if cmd cmplt is more than seven cmd_cmplt, and cmd cmplt is not cleared, cmd_cmplt_cnt overflows.

**Table 21-33. Command Queue Flush Register (Offset = 0x0208)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | - | - | Reserved |
| [30:28] | cmd_cmplt_cnt_ch7 | R | Command complete counter of Channel 7 |
| 27 | - | - | Reserved |
| [26:24] | cmd_cmplt_cnt_ch6 | R | Command complete counter of Channel 6 |
| 23 | - | - | Reserved |
| [22:20] | cmd_cmplt_cnt_ch5 | R | Command complete counter of Channel 5 |
| 19 | - | - | Reserved |
| [18:16] | cmd_cmplt_cnt_ch4 | R | Command complete counter of Channel 4 |
| 15 | - | - | Reserved |
| [14:12] | cmd_cmplt_cnt_ch3 | R | Command complete counter of Channel 3 |
| 11 | - | - | Reserved |
| [10:8] | cmd_cmplt_cnt_ch2 | R | Command complete counter of Channel 2 |
| 7 | - | - | Reserved |
| [6:4] | cmd_cmplt_cnt_ch1 | R | Command complete counter of Channel 1 |
| 3 | - | - | Reserved |
| [2:0] | cmd_cmplt_cnt_ch0 | R | Command complete counter of Channel 0 |

### 21.5.25 Command Complete Counter Reset (Offset = 0x020C)

The command completion counter can be reset by writing 1 to this register.

**Table 21-34. Command Queue Flush Register (Offset = 0x020C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [7:0] | cmd_cmplt_cnt_rst | W | Command complete counter reset<br>Write 1: Clear the counter of the corresponded channel<br>Write 0: No effect |

### 21.5.26 Command Format

Please refer to Section 4.2.28 for more details.

### 21.5.27 General Command Queue (Offset = 0x0280 ~ 0x0294)

Writing to this register will push the command to all command queues. Reading from this register will get all zero returns. Please note that this register cannot be set if one or more command queues are full. The general command queue can only be pushed with a command containing no data transfer from or to BMC (For example: when block erase, reset, and spare read/write are acceptable).

### 21.5.28 Command Queue (Offset = 0x0300 + *n* * 20 ~ 0x0314 + *n* * 0x20, *n* = 0 ~ 7)

The operation of NANDC is decided by the command pushed into the command queue. Each command contains six words and each command queue may have two or four space entries for storing the command. A command will be pushed into the command queue by writing the sixth word of a command and will pop from the command queue when it is completely executed by a NAND channel controller.

During reading the command queue register, the returned data is the currently processed (By NANDC) command, but not the current command to be written to the command queue.

**Table 21-35. Command Queue First Word (Offset = 0x0300 + *n* * 0x20, *n* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:30] | inc_ce_val | - | CE incremented value<br>This register decides the values of the CE increments when performing at the inc_ce state in a flow control.<br>2'b00: CE increases 1.<br>2'b01: CE increases 2.<br>2'b10: CE increases 3.<br>2'b11: Reserved |

| Bit | Name | Type | Description |
|---|---|---|---|
| [29:27] | - | - | Reserved |
| 26 | auto_ce_factor | R/W | Automatic CE selection multiply factor<br>1: Multiplied by 2<br>0: Multiplied by 1 (Original automatic CE selection value) |
| 25 | auto_ce_sel | R/W | Automatic CE selection<br>1: Enable the automatic CE selection mechanism<br>0: Disable the automatic CE selection mechanism<br>Note: If auto_ce_sel is enabled, fill the start ce function is not useful. If auto_ce_sel is enabled, start CE is decided by 1st_row_addr, which truncates the dv_size bits. |
| 24 | icwc | R/W | Internal copy back with correction<br>1: Enable<br>0: Disable<br>Note: When the icwc and ignore modes are enabled and the page size is the large page (2K, 4K, or 8K), the command index can set the ICWC fixed flow; otherwise, the ICWC fixed flow is prohibited. |
| [23:0] | row_addr_1st | R/W | First row address/source row address (Page index)<br>The first row address decides the starting row address at auto_ce_sel enable or the source address in a copy-back or a blanking check flow. |

**Table 21-36. Command Queue Second Word (Offset = 0x0304 + *n* * 0x20, *n* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:30] | dec_ce_val | - | CE decremented value<br>This register decides the value of the CE decrements when performing at the dec_ce state in a flow control.<br>2'b00: CE decreases 1.<br>2'b01: CE decreases 2.<br>2'b10: CE decreases 3.<br>2'b11: Reserved |
| [29:24] | - | - | Reserved |
| [23:0] | row_addr_2nd | R/W | Second row address/Source row address (Page index)<br>The second row address decides the second chip row address in a interleaving flow, or decides the destination address in a copy-back or blanking check flow. |

**Table 21-37. Command Queue Third Word (Offset = 0x0308 + *n* * 0x20, *n* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | reload_val | R/W | Reload the counter at the cache read/write or cache copy back fixed flow |
| [23:0] | row_addr_3rd | R/W | Third row address<br>The third row address decides the third chip row address in an interleaving flow. |

**Table 21-38. Command Queue Fourth Word (Offset = 0x030C + *n* * 0x20, *n* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | - | - | Reserved |
| [23:0] | row_addr_4th | R/W | Fourth row address<br>The fourth row address decides the fourth chip row address in an interleaving flow. |

**Table 21-39. Command Queue Fifth Word (Offset = 0x0310 + *n* * 0x20, *n* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | counter | R/W | Counter (For repeated operations)<br>If the counter is used in a data transfer flow, the scale will show as below:<br>If the ECC base is set to 1 kB, each count will represent 1 kB of data.<br>If the ECC base is set to 512 bytes, each count will represent 512 bytes of data.<br>Please refer to Table 5-3 for more detailed information. |
| [15:8] | sec_offset_2nd | R/W | The second sector offset<br>In the byte mode, this register decides the second column address (Issued with the first row address).<br>Note: This value is only effective in the byte mode. |
| [7:0] | sec_offset_1st | R/W | The 1st sector offset<br>This register decides the sector offset and hardware decode of this register to decide the first column address.<br>In the byte mode, this register decides the first column address (Issued with the first row address). |

**Table 21-40. Command Queue Sixth Word (Offset = 0x0314 + *n* * 0x20, *n* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:30] | flash_type | R/W | Support Flash Operation type<br>2'b00: Legacy Flash<br>2;b01: Reserved<br>2'b10: Toggle Flash<br>2'b11: ONFI Flash |
| [29:27] | start_ce | R/W | Command starting CE<br>This register decides the Flash from which the current command starts.<br>3'b000: CE0<br>3'b001: CE1<br>3'b010: CE2<br>3'b011: CE3<br>3'b100: CE4<br>3'b101: CE5<br>3'b110: CE6<br>3'b111: CE7 |

| Bit | Name | Type | Description |
|---|---|---|---|
| 26 | byte_mode | R/W | Byte mode<br>In the byte more, the spare SRAM is updated from the SRAM byte offset 0 of the corresponded region 0. The selection of the spare SRAM access region has no effect.<br>1: Enable<br>0: Disable<br>Note: spare_access_mode = 0, byte_mode = 0 is allowed.<br>spare_access_mode = 0, byte mode = 1 is allowed.<br>spare_access mode = 1, byte mode = 0 is allowed.<br>spare access mode = 1, byte mode = 1 is not allowed. |
| 25 | bmc_ignore | R/W | Ignore the BMC region status of full/empty (User mode)<br>1: Ignore<br>0: Do not ignore<br>Please refer to Chapter 5 for more detailed information. |
| [24:22] | bmc_region_sel | R/W | BMC region selection<br>3'b000 ~ 3'b111: Region 0 ~ Region 7<br>Note: The NAND channels 0 ~ 3 can only select regions 0 ~ 3 and the NAND channels 4 ~ 7 can only select regions 4 ~ 7.<br>Please refer to Chapter 5 for more detailed information. |
| [21:18] | spare_num | R/W | Number of spare date byte<br>Please refer to Table 4-40 for more detailed information. |
| [17:8] | cmd_index | R/W | Command index selection<br>These bits decide the starting position of the control flow.<br>Please refer to Table 5-3 for more detailed information. |
| 7 | pg_flow_sel | R/W | Program flow selection<br>1: Programmable control flow<br>0: Fixed Flow<br>Note: Users can program the micron code when the programming flow control register (0x600) starts. When setting pg_flow_sel to '1', the controller will execute the programming flow at the programming flow control register; otherwise, the controller will execute the fixed flow at the default ROM code. |
| [6:5] | spare_sram_sel | R/W | Spare SRAM access region selection<br>2'b00: Region 0<br>2'b01: Region 1<br>2'b10: Region 2<br>2'b11: Region 3<br>If the command queue depth is configured to '2', only region 0 and region 1 will exist and the legal value of this register will be '00' and '01'. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 4 | cmd_hsk_mode | R/W | Command handshake mode (Use the DMA handshake mode with the source DMA)<br>1: Enable<br>0: Disable |
| [3:2] | cmd_scale | R/W | Command incremental scale<br>2'b11: Reserved<br>2'b10: By two block<br>2'b01: By one block<br>2'b00: By page |
| 1 | - | - | Reserved |
| 0 | cmplt_intr_en | R/W | Command complete interrupt and status enable<br>1: Enable<br>0: Disable<br>Note: When cmplt_intr_en is set, users must clear the command complete interrupt and the host controller will execute next command at the command queue. Otherwise, if disable cmplt_intr_en has no command complete interrupt occurred, host will execute next command automatically. Users should poll the command complete counter to know which command is executed. |

**Table 21-41. Spare Number Usage**

1. Sector Mode

X=page_size/ecc_base
(unit is sector)

| X | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|
| max:spare_num unit(byte) | 4 | 8 | 16 | 16 | 16 |

ex: page size=4k, ecc base is 512byte, X=4k/512=8, so spare data length is from 1~8
spare_num is 0~7

2. Page Mode
spare length is 4,8,16,32,64 byte

ex: whatever page size is, spare user data length can be choose any of 4, 8, 16, 32, 64 byte
so spare_num is 0, 1, 3, 7, 15

spare_num is 0=> spare user data length is 4 byte
spare_num is 1=> spare user data length is 8 byte
spare_num is 3=> spare user data length is 16 byte
spare_num is 7=> spare user data length is 32 byte
spare_num is 15=> spare user data length is 64 byte

**NOTE1:** High Speed mode(ONFI or TOGGLE mode),
spare data length must be even number because column address must be even

**NOTE2:** At byte Mode, spare user data length is from 1~16 byte(spare_num is from 0~15)

### 21.5.29 BMC Region Status Register (Offset = 0x0400)

This register provides the full or empty status for each region. The region halt status can also be read from this register (Region halt occurs when an ECC uncorrectable error is encountered and this region is not in the user mode). Once a region is halted, the region software reset will be required for solving the halt state. Except the FIFO region, each region contains a small buffer to temporarily keep the write data. Please note that the region buffer empty status must be '1' and the region full status must be '0' before setting the DMA Mode Write Data Fill Register (Offset = 0x0424).

**Table 21-42. BMC Region Status Register (Offset = 0x0400)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | region_buf_empty | R | Region buffer empty status<br>1: Region buffer is empty.<br>0: Region buffer is not empty.<br>(Bit 31 ~ Bit 24: Region 7 ~ Region 0) |
| [23:16] | region_halt | R | Region halt status<br>1: Region is halted.<br>0: Region is not halted.<br>(Bit 23 ~ Bit 16: Region 7 ~ Region 0) |
| [15:8] | region_full | R | Region full status<br>1: Region is full.<br>0: Region is not full.<br>(Bit 15 ~ Bit 8: Region 7 ~ Region 0) |
| [7:0] | region_empty | R | Region empty status<br>1: Region is empty.<br>0: Region is not empty.<br>(Bit 7 ~ Bit 7: Region 7 ~ Region 0) |

### 21.5.30 Region *n* User Mode Pointer Adjustment Register (Offset = 0x0404 + *n* * 0x4, *n* = 0 ~ 7)

When the ignore bit is set in a command, the operation will enter the user mode. The BMC region is used as a memory, not a FIFO. By setting this register, the read and write positions can be adjusted. In the user mode, one BMC region contains 16 grids. The size of each grid is 512 bytes. The pointers should be adjusted to the correct position every time before issuing a command in the user mode. When processing a command in the user mode with 1 kB per sector, the pointer must be set to an even value (Such as 0, 2, 4, or 8).

Because each region only contains 8K bytes space, the length of one read/write operation in the user mode should be keep to less than 8K bytes (Pointer offset + read/write sector count*n ≤ 16, n = 1 for 512 byte per sector; n = 2 for 1 kB per sector).

**Table 21-43. Region *n* User Mode Pointer Adjustment Register (Offset = 0x0404 + *n* * 0x4, *n* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:12] | - | - | Reserved |
| [11:8] | region_um_wrptr | W | Region *n* write pointer in the user mode (For the NAND Flash read command) |
| [7:4] | - | - | Reserved |
| [3:0] | region_um_rdptr | W | Region *n* read pointer in the user mode (For the NAND Flash write command) |

### 21.5.31 DMA Mode Write Data Fill/Read Data Pop Register (Offset = 0x0424)

By writing to this register once (Write '1' and cleared by itself), one 512-byte data (Dummy data) will be filled into or popped from the specified region. This is useful to write or read 512-byte data in Flash with 1-kB sector. The pop or fill operation must be executed only once before starting the read/write data movements in the DMA mode or after transferring all read/write data to/from AHB.

Region_buf_empty should be '1' and region_full should be '0' before setting the fill_data register. Fill_data should not be set while executing the NAND Flash read command.

Region_empty should be '0' before setting the pop_data register. Pop_data should not be set while executing the NAND Flash write command.

**Table 21-44. DMA Mode Write Data Fill/Read Data Pop Register (Offset = 0x0424)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | pop_data | W | This signal pops 512 bytes of data from specific region (Used only in the DMA mode).<br>Write 1: Pop 512 bytes of data<br>Write 0: No effect<br>(Bit 15 ~ Bit 8: Region 7 ~ Region 0) |
| [7:0] | fill_data | W | This signal fills 512 bytes of data into specific region (Used only in the DMA mode).<br>Write 1: Fills 512 bytes of data<br>Write 0: No effect<br>(Bit 7 ~ Bit 0: Region 7 ~ Region 0) |

## 21.5.32 Region Software Reset Register (Offset = 0x0428)

The region can be reset by writing to this register.

**Table 21-45. Region Software Reset Register (Offset = 0x0428)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | region_sw_rst | W | Region software reset (This reset is cleared by itself.)<br>Write 1: Reset<br>Write 0: No effect<br>(Bit 7 ~ Bit 0: Region 7 ~ Region 0) |

## 21.5.33 Force Region Fill Read Data Register (Offset = 0x042C)

By setting this register to '1', the region goes to a non-empty state and returns meaningless data to a requested AHB Master. This register can only be set when a region is not in the DMA handshake mode. (A region in the DMA handshake mode means that the command processed by a NANDC enables the DMA handshake mode).

This register is useful in solving the condition in which a DMA Master reads data from a region when an ECC uncorrectable event happens. Because the uncorrectable data cannot be read from this region, this bit must be set to prevent the DMA Master from hanging.

**Table 21-46. Force Region Fill Read Data Register (Offset = 0x042C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | force_fill_rd | R/W | Force region to fill read data<br>1: Enable<br>0: Disable<br>(Bit 7 ~ Bit 0: Region 7 ~ Region 0) |

### 21.5.34 Region *x* Remaining Sector Count of Read Data (Offset = 0x0430 + *x* * 0x4, *x* = 0 ~ 7)

This register represents the data remained in one region. The counting unit is in Sector (512 bytes or 1K bytes). Once an ECC uncorrectable error happens, the corresponding region will be halted in the uncorrected sector. The remaining sector counts can be read from this register.

**Table 21-47. Region *x* Remaining Sector Count of Read Data (Offset = 0x0430 + *x* * 0x4, *x* = 0 ~ 7)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:17] | - | - | Reserved |
| [16:0] | regionx_rd_cnt | R | Remaining sector counts of the read data in region *x* |

### 21.5.35 Revision Number Register (Offset = 0x0500)

**Table 21-48. Revision Number Register (Offset = 0x0500)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | rev_num | R | Revision number of FTNANDC023 |

### 21.5.36 Feature1 Register (Offset = 0x0504)

**Table 21-49. Feature Register (Offset = 0x0504)**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | HS_IF | R | High-speed interface selection<br>1: Disable HS_IF<br>0: Enable HF_IF |

| Bit | Name | Type | Description |
|---|---|---|---|
| 30 | ahb_s3_clk_mode | R | AHB Slave3 interface clock mode selection<br>1: Asynchronous mode<br>0: Synchronous mode |
| 29 | ahb_s2_clk_mode | R | AHB Slave2 interface clock mode selection<br>1: Asynchronous mode<br>0: Synchronous mode |
| 28 | ahb_s1_clk_mode | R | AHB Slave1 interface clock mode selection<br>1: Asynchronous mode<br>0: Synchronous mode |
| 27 | ahb_s0_clk_mode | R | AHB Slave0 interface clock mode selection<br>1: Asynchronous mode<br>0: Synchronous mode |
| 26 | rgf_clk_mode | R | Register interface clock mode selection<br>1: Asynchronous mode<br>0: Synchronous mode |
| 25 | hs_clk_mode | R | High-speed NAND interface clock mode selection<br>1: The mode is selected by the port.<br>0: Synchronous mode |
| 24 | ns_clk_mode | R | Normal-speed NAND interface clock mode selection<br>1: The mode is selected by the port.<br>0: Synchronous mode |
| 23 | ecc_cs_unfold | R | Unfold value of the ECC Chien search correction engine<br>1'b0: 4<br>1'b1: 8 |
| [22:20] | ecc_cs_num | R | Number of ECC Chien search correction engine<br>3'b000: 1<br>3'b001: 2<br>3'b010: 3<br>3'b011: 4<br>3'b100: 5<br>3'b101: 6<br>Others: Reserved |
| [19:12] | pfc_depth | R | Programmable flow control register depth<br>Possible values: 32, 64, and 128 |

| Bit | Name | Type | Description |
|---|---|---|---|
| [10:8] | cmdq_depth | R | Command queue depth<br>Possible value: 2 or 4 |
| [7] | icwc_on | R | ICWC function turn-on control<br>1: Turn on the ICWC function (The ICWC flow can be executed.)<br>0: Turn off the ICWC function (The ICWC flow cannot be executed.) |
| [6:4] | ahb_slave_num | R | Number of the AHB data slave ports<br>Possible value: 1, 2, 3, or 4 |
| [3:0] | nandc_ch_num | R | Number of the NANDC channels<br>Possible value: 1, 4, or 8 |

### 21.5.37 AHB Slave Memory Space Range Register (Offset = 0x0508)

Each AHB slave data port contains eight memory spaces corresponded to each region. The range of the memory space can be set in this register from 512 bytes to 64K bytes.

**Table 21-50. AHB Slave Memory Space Range Register (Offset = 0x0508)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:25] | - | - | Reserved |
| [26] | split_en | R/W | AHB bus split protocol enable<br>1: Enable<br>0: Disable<br>Note: All slave ports use the same setting. |
| [24:16] | ahb_rd_len | R/W | AHB slave read data prefetch length<br>To improve the performance of a read operation, the read data can be pre-fetched with a length set in this register.<br>9'h1 ~ 9'h100: 1 ~ 256 words<br>Other value is not allowed.<br>Note: It is suggested setting this value to power of 2. For example, set this value to 64 or 128 words. |

| Bit | Name | Type | Description |
|---|---|---|---|
| [15:12] | ahb_force_len | R/W | Enable the read pre-fetch by the length set in ahb_rd_len<br>If this bit is set as '1', the AHB slave port will pre-fetch the length specified in "ahb_rd_len". If this bit is set as '0', the pre-fetched length will be decided by HBUSRT at each AHB transaction.<br>1: Pre-fetch length depends on ahb_rd_len.<br>0: Pre-fetch length depends on HBURST.<br>(Bits[15:12] are corresponding to AHB Slave 3 to 0, respectively.) |
| [11:8] | ahb_retry_en | R/W | AHB bus retry protocol enable<br>1: Enable<br>0: Disable |
| [7:0] | ahb_mem_range | R/W | AHB slave memory space range<br>8'b0000_0001: 512 bytes<br>8'b0000_0010: 1K bytes<br>8'b0000_0100: 2K bytes<br>8'b0000_1000: 4K bytes<br>8'b0001_0000: 8K bytes<br>8'b0010_0000: 16K bytes<br>8'b0100_0000: 32K bytes<br>8'b1000_0000: 64K bytes |

### 21.5.38 Global Software Reset Register (Offset = 0x050C)

Writing '1' to this register resets all modules in FTNANDC023, except the asynchronous AHB data salve and the following registers.

- ECC control-related register
- NANDC control-related register ("NANDC Auto Compare Pattern Register" is cleared.)
- Programmable flow control register
- Programmable OP-code register

After setting these registers, the NANDC software reset register must be '0' before entering further operations.

**Table 21-51. Global Software Reset Register (Offset = 0x050C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:1] | - | - | Reserved |
| 0 | glb_sw_rst | W | Global software reset<br>Write 1: Reset<br>Write 0: No effect |

## 21.5.39 AHB Data Slave Reset Register (Offset = 0x0510)

**Table 21-52. Global Software Reset Register (Offset = 0x0510)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:4] | - | - | Reserved |
| [3:0] | ahb_rst | R/W | AHB data slave software reset<br>This reset is only effective for the asynchronous AHB data slave port. After setting this field to '1', F/W must wait until this field is cleared to '0' by H/W. It should be noted that when the AHB slave port is in the synchronous mode, this register should not be written.<br>Write 1: Reset<br>Write 0: No effect |

## 21.5.40 ECC Correction Capability Register (Offset = 0x0514)

**Table 21-53. Feature Register (Offset = 0x0514)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | - | - | Reserved |
| 23 | ECC_CORR_24BIT_EN | R | ECC correction 24 bit |
| 22 | ECC_CORR_23BIT_EN | R | ECC correction 23 bit |
| 21 | ECC_CORR_22BIT_EN | R | ECC correction 22 bit |
| 20 | ECC_CORR_21BIT_EN | R | ECC correction 21 bit |
| 19 | ECC_CORR_20BIT_EN | R | ECC correction 20 bit |
| 18 | ECC_CORR_19BIT_EN | R | ECC correction 19 bit |
| 17 | ECC_CORR_18BIT_EN | R | ECC correction 18 bit |
| 16 | ECC_CORR_17BIT_EN | R | ECC correction 17 bit |
| 15 | ECC_CORR_16BIT_EN | R | ECC correction 16 bit |
| 14 | ECC_CORR_15BIT_EN | R | ECC correction 15 bit |
| 13 | ECC_CORR_14BIT_EN | R | ECC correction 14 bit |
| 12 | ECC_CORR_13BIT_EN | R | ECC correction 13 bit |
| 11 | ECC_CORR_12BIT_EN | R | ECC correction 12 bit |
| 10 | ECC_CORR_11BIT_EN | R | ECC correction 11 bit |
| 9 | ECC_CORR_10BIT_EN | R | ECC correction 10 bit |
| 8 | ECC_CORR_9BIT_EN | R | ECC correction 9 bit |
| 7 | ECC_CORR_8BIT_EN | R | ECC correction 8 bit |
| 6 | ECC_CORR_7BIT_EN | R | ECC correction 7 bit |
| 5 | ECC_CORR_6BIT_EN | R | ECC correction 6 bit |
| 4 | ECC_CORR_5BIT_EN | R | ECC correction 5 bit |
| 3 | ECC_CORR_4BIT_EN | R | ECC correction 4 bit |
| 2 | ECC_CORR_3BIT_EN | R | ECC correction 3 bit |
| 1 | ECC_CORR_2BIT_EN | R | ECC correction 2 bit |
| 0 | ECC_CORR_1BIT_EN | R | ECC correction 1 bit |

### 21.5.41 ECC Correction Capability Register (Offset = 0x0518)

**Table 21-54. dqs_in_delay Register (Offset = 0x0518)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:5] | - | - | Reserved |
| [4:0] | R_dqs_delay | R | Read DQS delay counter<br>Note: This register will exist when FPGA is set to ON.<br>0: Not delayed<br>1: Delayed for 1 T |

### 21.5.42 Programmable Flow Control Register (Offset = 0x0600+*n* * 0x4, *n* = 0 ~ 31)

The programmable flow control sequence can be set by writing this register. The maximum available space for the programmable flow control is 128 bytes.

**Table 21-55. Programmable Flow Control Register (Offset = 0x0600+*n**0x4, *n* = 0 ~ 31)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | pfc_*x*+3 | R/W | Programmable flow control byte *x* + 3 (*x* = n * 4, n = 0 ~ 31) |
| [23:16] | pfc_*x*+2 | R/W | Programmable flow control byte *x* + 2 (*x* = n * 4, n = 0 ~ 31) |
| [15:8] | pfc_*x*+1 | R/W | Programmable flow control byte *x* + 1 (*x* = n * 4, n = 0 ~ 31) |
| [7:0] | pfc_*x* | R/W | Programmable flow control byte *x* (*x* = n * 4, n = 0 ~ 31) |

### 21.5.43 Programmable OP-code Register (Offset = 0x0700 ~ 0x0704)

The OP-code can be set by this register.

**Table 21-56. Programmable OP-code Register (Offset = 0x0700)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | opcode_3 | R/W | Programmable OP-code byte 3 |
| [23:16] | opcode_2 | R/W | Programmable OP-code byte 2 |
| [15:8] | opcode_1 | R/W | Programmable OP-code byte 1 |
| [7:0] | opcode_*0* | R/W | Programmable OP code byte *0* |

**Table 21-57. Programmable OP-code Register (Offset = 0x0704)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | opcode_*7* | R/W | Programmable OP-code byte 7 |
| [23:16] | opcode_*6* | R/W | Programmable OP-code byte 6 |
| [15:8] | opcode_*5* | R/W | Programmable OP-code byte 5 |
| [7:0] | opcode_*4* | R/W | Programmable OP code byte *4* |

### 21.5.44 Spare SRAM Access Register (Offset = 0x1000 ~ 0x2FFF)

The spare data can be read or written by accessing this register. The Flash ID is also read from this register. The spare SRAM format is described in Section 5.8 in details.

### 21.5.45 Data SRAM Access Register (Offset = 0x1_0000 ~ 0x1_1FFF)

The data in BMC FIFO can be read or written by accessing this register. The mapping of the address offset to the exact region position is depicted below.

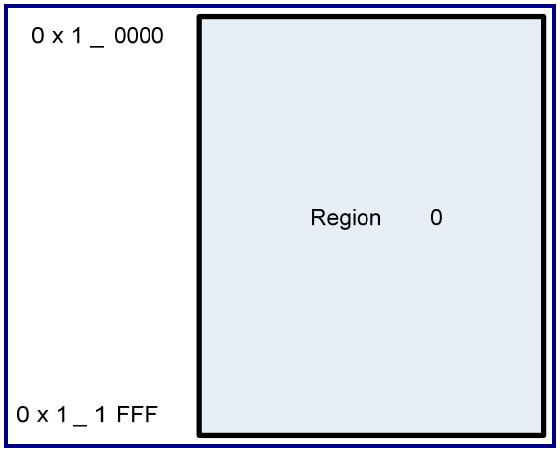



**Figure 21-18. Data SRAM Address Offset Mapped to BMC Region**

## 21.6 AHB Data Slave Port

FTNANDC023 comprises one data slave port and this data slave port has a memory space from 512 B to 64 kB.

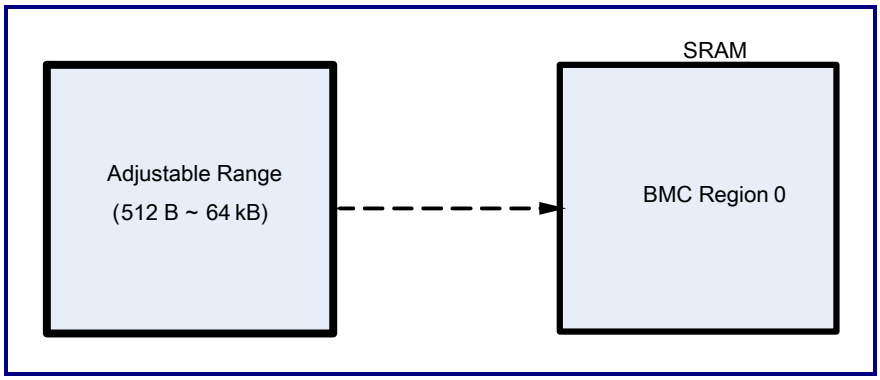



**Figure 21-19. Mapping Between AHB Slave and BMC Region 0**

The AHB data slave port supports the AHB retry protocol. When "ahb_retry_en" is set, the AHB data slave port will retry if the selected BMC region is not available for accessing.

## 21.7 Buffer Management Controller

The Buffer Management Controller (BMC) is used to control the data accessed from NANDC or AHB data slave port. The data storage space is divided into eight regions when the configured NANDC is eight, four regions when the configured NANDC is four, and one region when the configured NANDC is one).

The field, "bmc_region_sel", in the sixth word of a NANDC command decides the region to be accessed during processing a NANDC command. Please note that the NANDC channels 0 to 3, can only access the BMC regions 0 to 3, and the NANDC channels 4 to 7 can only access the BMC regions 4 to 7.

Setting the "bmc_ignore" bit in the sixth word of a NANDC command to '1' indicates that this command is in the user mode. Read/Write from NANDC does not affect the region status used in the normal mode when operating in the user mode. The user mode pointer can be adjusted by writing "User Mode Pointer Adjustment Register". The following example shows the read operation in the user mode from the NANDC channel 0 to region 0. Please note that the data are ready in region 0 only after the NAND command is completed and ECC is enabled.

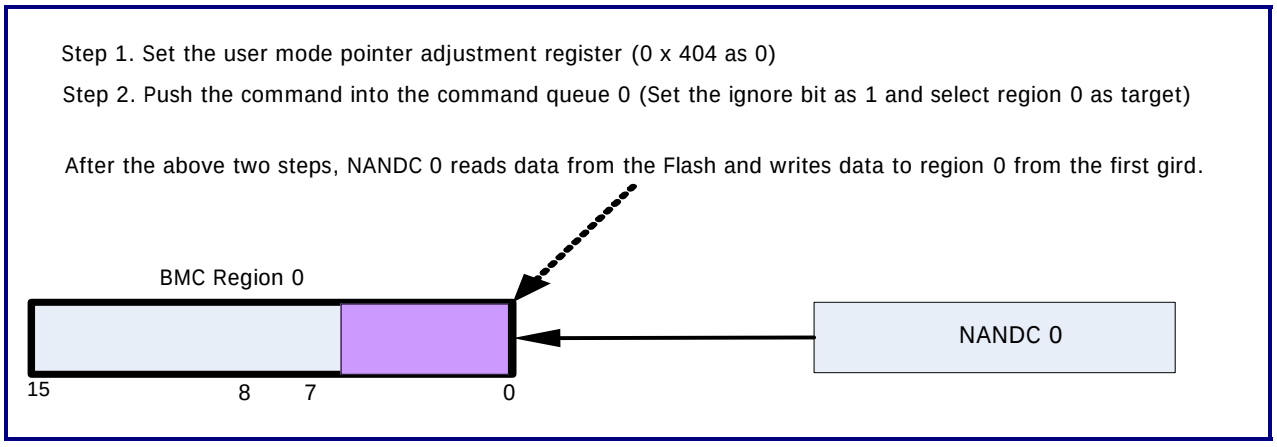



**Figure 21-20. User Mode Operation**

When operating in the normal mode (That is, the NANDC command with the "bmc_ignore" bit is '0'), the region acts as FIFO. For a Flash write operation, data are pushed from the AHB data slave port, and NANDC pops the data from the designated region. For a Flash read operation, data are pushed from NANDC, and is popped by the AHB data slave port after the ECC checking and correction.

The DMA handshake protocol is supported in FTNANDC023. When a NANDC command is issued and "cmd_hsk_mode" (In the sixth word of a NANDC command) is set as '1', the dma_req signal will be asserted in the following conditions:

- For a write operation, the dma_req signal is asserted when at least a space of 512 bytes is available in a region.
- For a read operation, the dma_req signal asserts when at least one sector can be read by the AHB data slave port. Each BMC region has one dedicated dma_req/dma_ack I/O, that is, bit 0 to bit 7 of dma_req/dma_ack correspond to region 0 to region 7 of BMC, respectively. The transfer length of one pair of dma_req/dma_ack (DMA burst size) is 512 bytes of data.

Please note that the designated BMC region must not conflict with different NAND channels when issuing a NAND command. For example, if NAND channel 0 designates region 0 as a target, NAND channel 1 must not designate region 0 as a target before NAND channel 0 has finished the operation.

## 21.8 AHB Register Slave Port

FTNANDC023 comprises one AHB register slave port. This port is used to access the following items:

1. FTNANDC023 register
2. Command queue
3. Programmable flow control register
4. Programmable OP-code register
5. Spare SRAM
6. Data SRAM

Please note that HSIZ of the command queue can only be accessed with WORD. The AHB register slave port can be configured as the asynchronous mode or synchronous mode. If configured as the synchronous mode, the AHB clock will be the same as the system clock. If configured as the asynchronous mode, the AHB clock will not have relationship with the system clock.

## 21.9 ECC Correction Error Handling

When an ECC correction fail event is encountered, the ECC correction fail interrupt status will be checked by reading the register offset 0x0024. Consequently, the failed channel will be halted; that is, the corresponding NAND channel controller will stop further operations. However, the designated BMC region may still contain correct sectors that can be read out. Once the correct sectors in the BMC region are all read out, the "BMC region halt" and "region_buf_empty" statuses will be set to '1' to be checked by reading the offset 0x0400.

The example procedure below is dealing with the ECC correction fail event:

1. Check "ECC Interrupt Status Register" (Offset = 0x0024) to identify the ECC correction failed channel
2. Poll "BMC Region Status Register" (Offset = 0x0400) to make sure that the correct data in the designated region are all read out.
3. Check "Region x Remaining Sector Count of Read Data" (Offset = 0x0430) with values obtained in this register and row address in the command queue, the precise page location of the ECC uncorrectable error can be calculated.
4. Reset the designated BMC region

5. Flush the corresponded command queue (This step is optional. If the command queue is not flushed, NANDC will repeat the current command after reset.)
6. Poll "0x100" to check if the target channel is at the ready state.
7. Reset NANDC (This step can be skipped if Step 5 is executed.)
8. Poll the NANDC software reset register until it returns to '0'.

## 21.10 Auto Compare Error Handling

When the auto compare fail event occurs during manipulating the blanking check, command complete will not be asserted, and F/W read register (0x0154) will find auto compare fail. The host controller will hang and F/W must perform the following sequence.

1. Check if the auto_cmp_fail (0x0154) status occurred.
2. Flush the command queue (Including resetting the controller and ECC engine)
3. Poll the NANDC software reset register until it returns to '0'

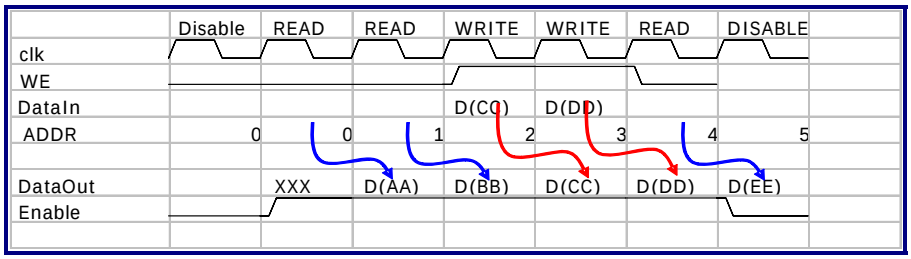
## 21.11 Abort Sequence

When the host controller wants to abort a command, the sequence below should be followed to guarantee a correct flow.

1. Flush the command queue
2. Poll the NANDC software and cleared to '0'
3. Reset the BMC Region
4. Reset the AHB Slave (If the AHB slave port is asynchronous.) and wait for the AHB reset to be cleared
5. Issue the "RESET FLASH" command before issuing a new command

## 21.12 Rom/SRAM Behavior

ROM/SRAM must be included in this IP, and the ROM/SRAM behavior must be the same as the following waveform to meet the design timing requirement.




Especially when users generate ROM/RAM at the FPGA stage, the ROM/RAM behavior must be the same as the above waveform.

## 21.13 Host Controller Data Format

### 1. Legacy Flash Operation

The ECC parity is combined with Flash data. The spare data and crc (sector_mode)/ecc_parity (page_mode) are using the data format to reduce the random data access time in order to improve the performance. Figure 5-3 depicts the data format of the host controller.

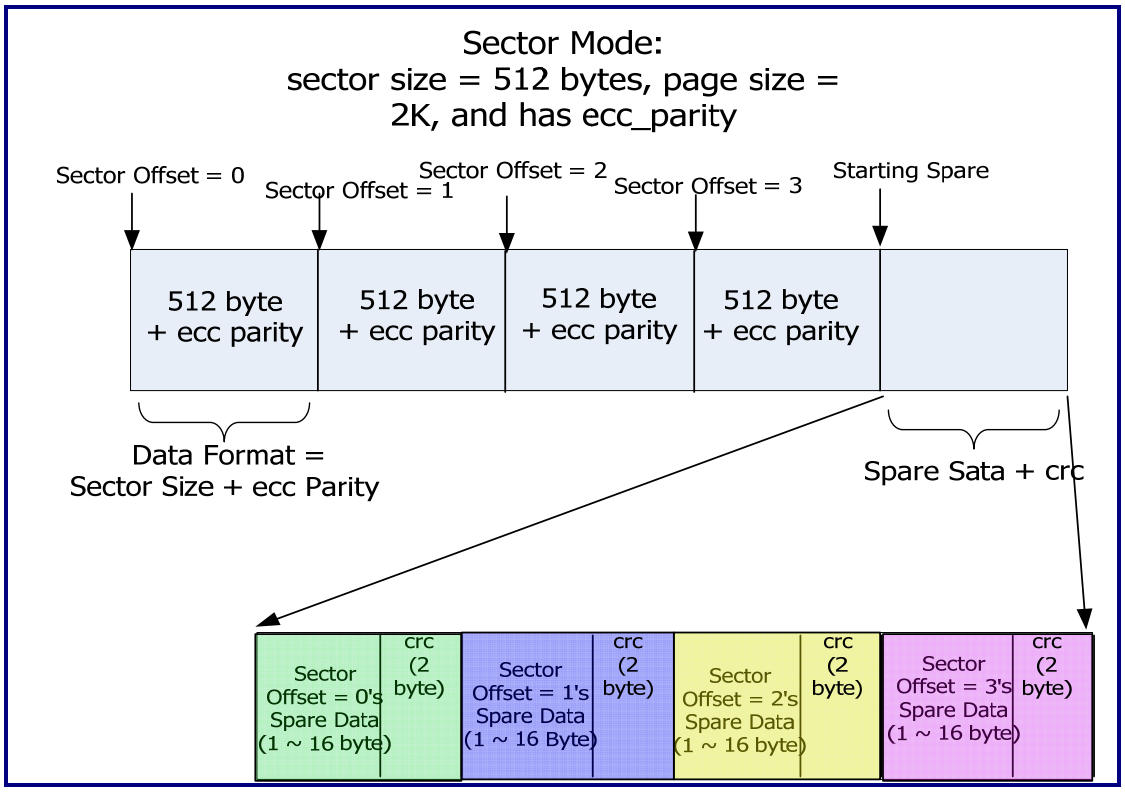



**Figure 21-21. Sector Mode Data Format**

The maximum spare data length is ranging from 1 byte to 16 bytes by programming the settings of page_size and ecc_base. When crc is disabled, the crc (2 byte) stage will not show.

**2. High-speed Flash Operation**

If F/W programs ecc_parity to be an odd number, the data strobe must be the even number specified in the high-speed Flash specification. Therefore, ecc_parity is automatically added by 1 by using the host controller that meets the even number of the high-speed Flash. Thus, the used spare data will be compressed. For example, if the page size is 4K, sector size is 512 bytes, spare data size is 128 bytes, and ECC correct is 6 bits (ecc_parity is 11 byte); there will be 4K/512byte = 8 sectors. The remained spare data size will be 128 - (8 * (11 + 1)) = 32 bytes at the high-speed Flash operation. However, for the legacy Flash operation, the remained spare data size will be 128 - (11 * 8) = 40 bytes.

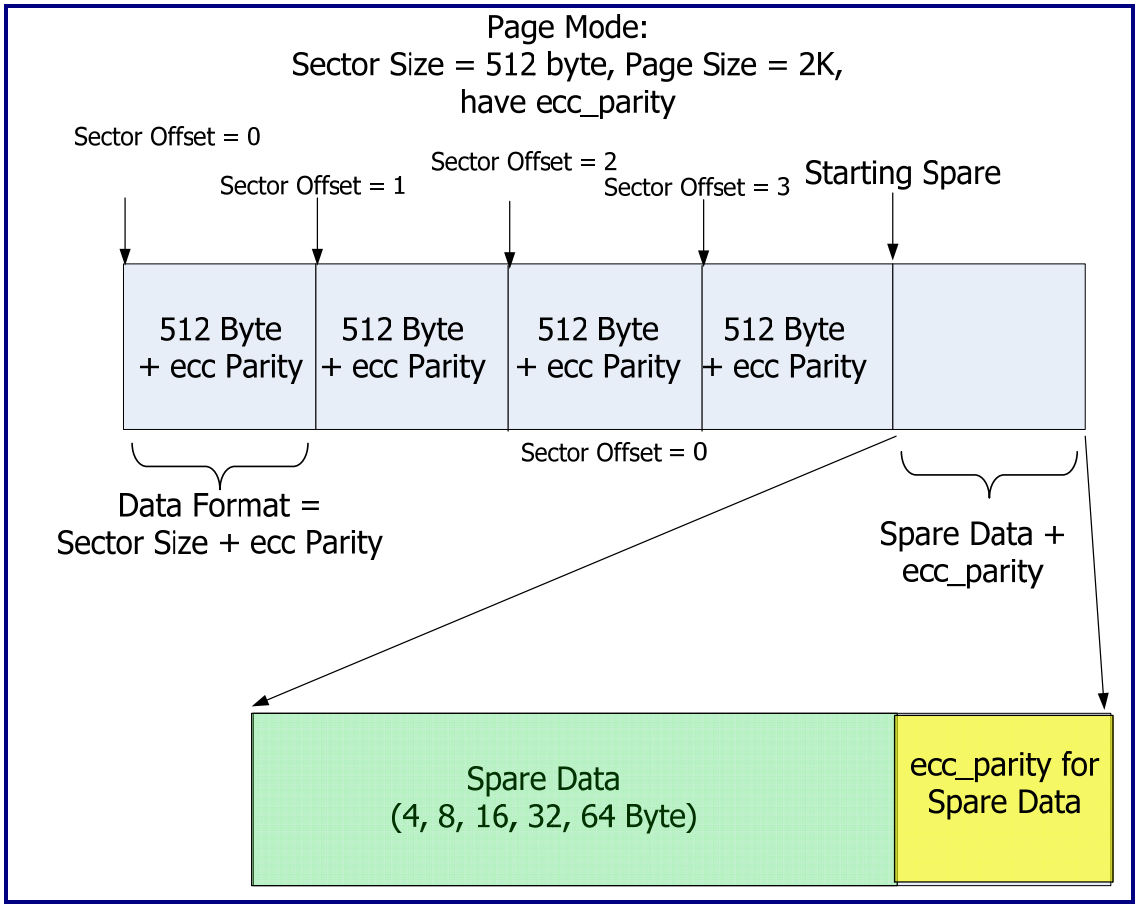



**Figure 21-22 Page Mode Data Format**

The user-defined data length can be 4, 8, 16, 32, or 64 bytes by programming the setting and disabling the spare region ECC; except that ecc_parity is still active.

## 21.14 Description of NANDC MicroCode

**Table 21-58. Summary of MicroCode**

| MircoCode | bit[7:0] | operation comment |
|---|---|---|
| **opcode** | 8'b00_?????? | bit[5:0]means different opcode command, at most 64 opcode |
| **zero_addr** | 8'b010_00000 | issue address with all zero value, cycle number is setting by col_cyc |
| **1st_row_addr** | 8'b010_00001 | 1st_row address cycles, cycle number is setting by row_cyc |
| **1st_col_addr** | 8'b010_00010 | 1st_col address cycles is setting by col_cyc, and colmun address is generated by hardware base on 1st_sec_offset |
| **2nd_row_addr** | 8'b010_00011 | 2nd_row address cycles is setting by row_cyc |
| **2nd_col_addr** | 8'b010_00100 | 2nd_col address cycles is setting by col_cyc, and colmun address is generated by hardware base on 2nd_sec_offset |
| **3rd_row_addr** | 8'b010_00101 | 3rd_row address cycles, and row cycle is setting by row_cyc |
| **4th_row_addr** | 8'b010_00110 | 4th_row_address cycles, and row cycle is setting by row_cyc |
| **tog_1st_addr** | 8'b010_00111 | toggle 1st_row_addr, and toggle address cycles is setting by row_cyc. toggle bit deponds on user setting |
| **tog_2nd_addr** | 8'b010_01000 | toggle 2nd_row_addr, and toggle address cycles is setting by row_cyc. toggle bit deponds on user setting |
| **tog_3rd_addr** | 8'b010_01001 | toggle 3rd_row_addr, and toggle address cycles is setting by row_cyc. toggle bit deponds on user setting |
| **tog_4th_addr** | 8'b010_01010 | toggle 4th_row_addr, and toggle address cycles is setting by row_cyc. toggle bit deponds on user setting |
| **sp_col_addr** | 8'b010_01011 | spare colmun address for flash random opcode . depends on 1st_sec_offset. col_cyc means source sp_col_addr cycles. |
| **fix_3rd_addr** | 8'b010_01100 | fix 3rd_row_addr, row_cyc is fix addr cycles. |
| **fix_4th_addr** | 8'b010_01101 | fix 4th_row_addr row_cyc is fix addr cycles. |
| **buffer1** | 8'b011_00000 | buffer1: max{tADL,tCCS}~60ns |
| **buffer2** | 8'b011_00001 | buffer2: max{tAR,tRR,tCLR}~20ns |
| **buffer3** | 8'b011_00010 | buffer3: max{tRHW,tRHZ}~100ns |
| **buffer4** | 8'b011_00011 | buffer4: max{tWHR}~60ns |
| **write data** | 8'b011_00100 | write data state including data+ecc_parity |
| **read data** | 8'b011_00101 | read data state including data+ecc_parity |
| **busy** | 8'b011_00110 | busy state means wait busy ready by hardware |
| **end** | 8'b011_00111 | control flow end state |
| **write spare** | 8'b011_01000 | write spare state |
| **read spare** | 8'b011_01001 | read spare state |
| **read status** | 8'b011_01010 | read status once |
| **multi read status** | 8'b011_01011 | read status until busy/ready OK |
| **blank write data** | 8'b011_01100 | write whole page data for blanking check using auto compare pattern. |
| **blank read data** | 8'b011_01101 | read whole page data for blanking check using auto compare pattern |
| **inc_ce** | 8'b011_01110 | inc ce state means increase ce |
| **dec_ce** | 8'b011_01111 | dec ce state means decrease ce |
| **reload** | 8'b011_10000 | reload r/w data and r/w spare counter for others data transfer |
| **DRV** | 8'b011_10001 | High Speed Driving State |
| **RETURN** | 8'b011_10010 | Return to Main Rountine from Sub-rountine |
| **goto** | 8'b10_?????? | Goto sun_rontine. bit[5:0] means goto sub_rountine index |
| **goback** | 8'b11_?????? | goback state means goback which index of control flow table. bit[5:0]means goback index number |

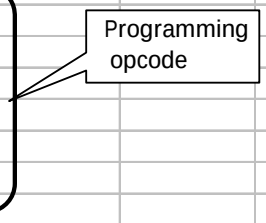
### 21.14.1 MicroCode Description

- opcode

Users can use the opcode index to choose the proper opcode based on Table 21-59.

**Table 21-59. opcode**

| opcode | | | | |
|---|---|---|---|---|
| 00 (index0) | 85 (index16) | FC (index32) | | |
| 05 (index1) | 8C (index17) | byte0 | | |
| 10 (index2) | 90 (index18) | byte1 | | |
| 11 (index3) | D0 (index19) | byte2 | | |
| 15 (index4) | E0 (index20) | byte3 | | |
| 30 (index5) | F1 (index21) | byte4 | | |
| 31 (index6) | F2 (index22) | byte5 | | |
| 35 (index7) | FF (index23) | byte6 | | |
| 3A (index8) | 01 (index24) | byte7 | | |
| 3F (index9) | 50 (index25) | | | |
| 60 (index10) | 8A (index26) | | | |
| 70 (index11) | 78 (index27) | | | |
| 71 (index12) | EC (index28) | | | |
| 7B (index13) | ED (index29) | | | |
| 80 (index14) | EE (index30) | | | |
| 81 (index15) | EF (index31) | | | |




For example, if users want to issue opcode of 10, which the index is 2, to fill 8'b000_00010.

- zero_addr:

  When users fill zero address in a flow, it indicates that the zero address is issued to Flash.

- 1st_row_addr:

  When users fill the first row address in a flow, it indicates that the host will issue the first-row address from the command table at non-auto ce sel. If auto ce sel is enabled, the address of $1^{st}$_row_addr[dv_size-1:0] will be issued. After the goback state, 1st_row_addr will be increased by using page or block to program the setting.

- 1st_col_addr:

  When it is not in the byte mode, the host will issue the first column address according to the first sec. offset in the command table. For example, if the first sec. offset is '1', the sector size will be 512 byte and the ECC parity will be 3 bytes. Then, the first column address can be calculated as (512 + 3). When it is in the byte mode, the first column address is concatenated with the second sec. offset and the first sec. offset becomes a 2-byte address, of which the address can be set to any value by users.

If the host is operating longer than the second time of 1st_col_addr in a flow, 1st_col_addr will become '0'.

- 2nd_row_addr:
  When users fill the second row address in a flow, it indicates that the host will issue the second row address from the command table. After the goback state, 2nd_row_addr will be increased using a page or block by programming the settings.
- 3rd_row_addr:
  When users fill the third row address in a flow, it indicates that the host will issue the third row address from the command table. After the goback state, 3rd_row_addr will be increased using a page or block by programming the settings.
- 4th_row_addr:
  When users fill the fourth row address in a flow, it indicates that the host will issue the fourth row address from the command table. After the goback state, 4th_row_addr will be increased using a page or block by programming the settings.
- toggle_addr:
  The host will issue an address to toggle the specified addresses. When toggling 1st_row_addr = 24'h0000_0000_0000, and the toggle bit is set to be 7, toggle_addr will be 24'h0000_0000_0080.
- sp_col_addr:
  The host will generate sp_col_addr according to the first sec. offset. For example, in the sector mode, the first sec. offset = 2, page size = 2k, sector_size = 512 bytes, ecc = 3 bytes, spare_num = 3 bytes of sector, and sp_col_addr = (512 + 3) * 4 + 3 * 2, the formula will be:
  *sp_col_addr = (sector_size + ecc_parity) * (page_size/sector_size) + (spare_num * 1st_sec_offset).*
  If the host has been operating longer than the second time of sp_col_addr in a flow, sp_col_addr becomes *(sector_size + ecc_parity) * (page_size/sector_size)*. In the page mode, sp_col_addr = (sector_size + ecc_parity) * (page_size/sector_size).
- fix_addr:
  The host will issue the fixed specific addresses.
- Buffer1:
  The buffer1 state issued by the host indicates that Flash will delay the AC timing by using the AC timing programming register.

- Buffer2:

  The buffer2 state issued by the host indicates that Flash will delay the AC timing by using the AC timing programming register.
- Buffer3:

  The buffer3 state issued by the host indicates that Flash will delay the AC timing by using the AC timing programming register.
- Buffer4:

  The buffer4 state issued by the host indicates that Flash will delay the C timing by using the AC timing programming register.
- write data:

  The host will perform the write data strobe and ECC parity strobe until the page boundary or sector counter is reached.
- read data:

  The host will perform the read data strobe and ECC parity strobe until the page boundary or sector counter is reached.
- busy:

  The host will wait for the Flash busy/ready until busy/ready is set to high.
- end:

  The host issues the end state to perform the end command of the Flash and the controller will finish this command.
- write spare:

  The host will issue proper write spare length according to the sector counter until the page boundary or sector counter is reached.
- read spare:

  The host will issue proper read spare length according to the sector counter until the page boundary or sector counter is reached.
- multi-read status:

  The host will pull the Flash status until this status is ready and the check status passes/fails.
- read status :

  The host will check the status once regardless the Flash is ready or busy.

- blank write data:

  The host will issue the blank write data strobe in a blanking flow and its length is the total page size (Including the spare area).
- blank read data:

  The host will issue the blank read data strobe in a blanking flow and its length is the total page size (Including the spare area).
- inc_ce:

  This state simply increases CE.
- dec_ce:

  This state simply decreases CE.
- reload:

  The is the reload counter for the write data, read data, write spare, and read spare operations. The reload value is recorded in the register files. This state can catch the read, write, or copy back flows.
- DRV:

  High-speed drive state (Reserved)
- Return:

  From subroutine "goback" to the main flow
- Goto:

  It goes from the main flow to the subroutine index. The Goto[5:0] index is the subroutine index.
- goback:

  The goback state will continually perform a flow if the total data performed exceed one page. Bit 5 to Bit 0 are the goback index numbers.

Note:

The goback state is 8'b11. Bit 5 to bit 0 represent the goback index number. If one flow includes a 10-microcode state and if the first state is opcode, the second state will be 1st_col address, and the 9$^{th}$ state will be goback. The state will go to the goback state with index = 8 and the current goback index = 9. Consequently, goback to 9 – 8 = 1, the next state is the opcode state.

## 21.14.2 Command Register Setting

Before executing a command, users must refer to the command register setting table to set the counter in the page/sector mode or to set in the byte mode.

**Table 21-60. Command Register Setting**

### Large-page fixed flow

| Coommand Register setting | cmd_index | bmc_ignore | byte_mode | cmd_hsk_mode | 1st_offset | 2nd_offset | counter unit at sector mode | counter unit at page mode | update spare |
|---|---|---|---|---|---|---|---|---|---|
| PAGE READ | 1B | V | 0 | V | V | don't care | sector | sector | X |
| PAGE WRITE WITH SPARE | 25 | V | 0 | V | V | don't care | sector | sector | V |
| SPARE WRITE | 30 | don't care | 0 | 0 | V | don't care | sector | page | V |
| SPARE READ | 3A | don't care | 0 | 0 | V | don't care | sector | page | X |
| PAGE READ with SPARE | 44 | V | 0 | V | V | don't care | sector | sector | X |
| READ ID (byte mode) | 4F | don't care | 1 | 0 | V | V | don't care | don't care | X |
| RESET | 55 | don't care | 0 | 0 | don't care | don't care | don't care | don't care | X |
| BLOCK ERASE | 58 | don't care | 0 | 0 | don't care | don't care | block | block | X |
| INTERNAL COPY BACK | 5F | don't care | 0 | 0 | V | don't care | sector | page | V |
| BLANKING CHECK | 6E | don't care | 0 | 0 | don't care | don't care | page | page | X |
| byte write(byte mode) | 7F | don't care | 1 | 0 | V | V | don't care | don't care | X |
| byte read(byte mode) | 88 | don't care | 1 | 0 | V | V | don't care | don't care | X |
| multi read status | 91 | don't care | 0 | 0 | don't care | don't care | don't care | don't care | X |
| read status | 93 | don't care | 0 | 0 | don't care | don't care | times/loop | times/loop | X |
| 2P PAGE WRITE/Samsang_Legacy | 96 | V | 0 | V | V | don't care | sector | sector | V |
| 2P PAGE WRITE/Micron | A9 | V | 0 | V | V | don't care | sector | sector | V |
| 2P BLOCK ERASE/Samsung_Legacy | BC | don't care | 0 | 0 | don't care | don't care | 2 block/loop | 2 block/loop | X |
| 2P PAGE READ | C5 | V | 0 | V | V | don't care | sector | sector | X |
| 2P/I2 PAGE WRITE/Samsung_Legacy | D9 | V | 0 | V | V | don't care | sector | sector | V |
| 2P COPY BACK/Samsung_Legacy | 101 | don't care | 0 | 0 | don't care | don't care | 2 page/loop | 2 page/loop | X |
| 2P COPY BACK/Intel | 114 | don't care | 0 | 0 | don't care | don't care | 2 page/loop | 2 page/loop | X |
| 2P/I2 BLK ERASE/Samsung_Legacy | 129 | don't care | 0 | 0 | don't care | don't care | 4 block/loop | 4 block/loop | X |
| 2P/I2 BLK ERASE/Toshiba/Samsung_1 | 13B | don't care | 0 | 0 | don't care | don't care | 4 block/loop | 4 block/loop | X |
| 2P PAGE WRITE/Samsang_TOG | 14D | V | 0 | V | V | don't care | sector | sector | V |
| 2 PLANE ERASE/Samsung_TOG | 160 | don't care | 0 | 0 | don't care | don't care | 2 block/loop | 2 block/loop | X |
| 2P COPY BACK/Samsang_TOG | 169 | don't care | 0 | 0 | don't care | don't care | 2 page/loop | 2 page/loop | X |
| 2P/I2 PAGE WRITE/Samsung_TOG | 17C | V | 0 | V | V | don't care | sector | sector | V |
| I2 PAGE READ | 1A4 | V | 0 | V | V | don't care | sector | sector | X |
| I2 PAGE WRITE | 1BE | V | 0 | V | V | don't care | sector | sector | V |
| I4 PAGE READ | 1D4 | V | 0 | V | V | don't care | sector | sector | X |
| I4 PAGRE WRITE | 206 | V | 0 | V | V | don't care | sector | sector | V |
| I4 BLOCK ERASE | 232 | don't care | 0 | 0 | don't care | don't care | 4 block/loop | 4 block/loop | X |
| I2 BLOCK ERASE | 24E | don't care | 0 | 0 | don't care | don't care | 2 block/loop | 2 block/loop | X |
| I2 HW COPY BACK | 25C | 1 | 0 | 0 | 0 | don't care | sector | sector | V |
| HW COPY BACK | 2A6 | 1 | 0 | 0 | 0 | don't care | sector | sector | V |
| HW COPY BACK CROSS CHIP | 2B9 | 1 | 0 | 0 | 0 | don't care | sector | sector | V |
| CACHE READ/Toshiba | 2D2 | V | 0 | V | 0 | don't care | sector | sector | X |
| CAHCE WRITE/Toshiba | 2E7 | V | 0 | V | 0 | don't care | sector | sector | V |
| COPY BACK WITH CAHCE/Toshiba | 2FC | don't care | 0 | 0 | 0 | don't care | 3 page/loop | 3 page/loop | X |
| INTERNAL COPY BACK with Correction | 324 | 1 | 0 | 0 | 0 | don't care | sector | sector | V |
| SYN RESET | 33A | don't care | 0 | 0 | don't care | don't care | don't care | don't care | X |
| READ PARAMETER PAGE | 33D | V | 0 | V | don't care | don't care | sector | sector | X |
| READ UNIQUE ID | 344 | V | 0 | V | don't care | don't care | sector | sector | X |
| GET FEATURE | 34B | don't care | 1 | 0 | don't care | don't care | don't care | don't care | X |
| SET FEATURE | 352 | don't care | 1 | 0 | don't care | don't care | don't care | don't care | X |
| SELECT LUN WITH STATUS | 358 | don't care | 0 | 0 | don't care | don't care | don't care | don't care | X |

Notes:

1: “don't care” indicates that users can fill with anything.

2: “V” indicates the value-affect behavior.

3: “2 block/loop” indicates that “loop” is the counter number if the counter is set to 2, which will be totally 2 * 2 = 4 blocks.

4: ecc_base(0x0008) = 1 indicates that one sector has 1024 bytes of data and ecc_base = 0 indicates that one sector has 512 bytes of data.

5: Counter unit at sector indicates that the counter unit is sector and the counter unit at page indicates that the counter unit is page.

6: Counter must be at the page boundary in the ICWC flow.

7: Copy back with cache: The counter unit of “3 page/loop” implies that one loop (Counter = 1) will copy at least three pages. If users want to copy five pages, the counter should be filled with 3 to copy five pages, which is given by: (counter + 2). If users want to copy eight pages, the counter should be filled with 6:.

8: Update spare writes the spare data to Flash from the spare SRAM.

9: Definitions of the counter units are different in the page mode and in the sector mode.

10: Cache read/write operation: The sector counter must be more than one page. Example 1: In cache write with 9 sectors, which is more than one page (8 sectors), counter (Command queue of fifth word) must be filled with 8 and reload_val (Command queue of third word) must be filled with 1; and totally to be 9 = 8 + 1. Example 2: In cache read with 16 sectors (Two pages), the counter must be filled with 8 and reload_val must be filled with 8; and totally to be 16 = 8 + 8. Example 3: In cache write with 25 sectors, the counter must be filled with 24 and reload_val must be filled with 1; and totally to be 25 = 24 + 1. Once reload_val represents the total sectors to be transferred minus multiple pages and reload_val must be filled with a value, zero in not allowed.

11: The number behind the command name is the command index. Users can fill in the specified value in the command table to perform a flow.

### 21.14.3 Flow Rules

The fixed flow and programming flow are comprised by MicoCode, as indicated in Table 5-1. The following rules must be followed.

- The counter unit can be sector, page, or block. Please refer to Command Register Setting (Table 5-3) (0 is inhibited).
- In the byte mode, only the read/write spare state can be used. The goback state cannot be in a flow. The sector counter can be ignored. spare_num is programmed from 0 to 15 (1 byte ~ 16 bytes).
- The row/column address is decided by the flow state. When the state goes to a flow state, the row address will be issued.
- In the byte mode, the first column address is decided by concatenating 1st_sec_offset and 2nd_sec_offset. 2nd_sec_offset is the high byte. When it is not in the byte mode, 1st_sec_offset and 2nd_sec_offset are the sector offsets. They can be programmed from 0 to 15. The first column address and the second column address will be decided.
- The sector format of the read/write data state is (Data + ecc_parity); however, ecc_parity will always appear whether or not ecc_en is enabled and no_ecc_parity is disabled. If no_ecc_parity is '1' and ecc_en is '0', ecc_parity will disappear in the data format.
- "intr_en" means that users must clear the command complete interrupt when a command finishes. The controller will then issue the next command if the command queue is not empty.
- The address will not be automatically generated across the chip boundary. This matter should be taken care by users.
- In a block erase flow, inc_by_page/blk must be set to in_by_blk.
- In a blanking check flow, in_by_page/blk must be set to in_by_page.
- The interleaving copy-back supports the same source and destination addresses in different chips.
- The interleaving/2-plane flow must be finished at the target Flash chip/plane. For example, when page_size = 2k, ecc_base = 512 bytes and users set the interleaving 2 flow from chip 0 to chip 1, the sector counter must be finished at chip 1. The 2-plane flow must be finished at plane 1.
- The command queue will pop when a command finishes normally. However, when auto compare fails or ECC fails, the command queue will not pop.
- The read status reads the Flash status once.
- The multiple read statuses read the Flash until the Flash is ready.

- The inc_ce and dec_ce states increase or decrease CE, which increases or decreases the value recoded in the command table.
- The blank write data/blank read data is used only in the blanking check flow. The total length = Page size + spare_size.
- 2-plane commands operate only in one block. For the next block, users must reissue a new command.
    - Example: As shown in Figure 21-23, block size = 64 pages. If users issue a 2-plane page write command to write four pages from page 63. This command will not be finished correctly by the host controller. Users must issue two commands; the 2-plane page writes to write two pages from page 63. Another command is from page 128.

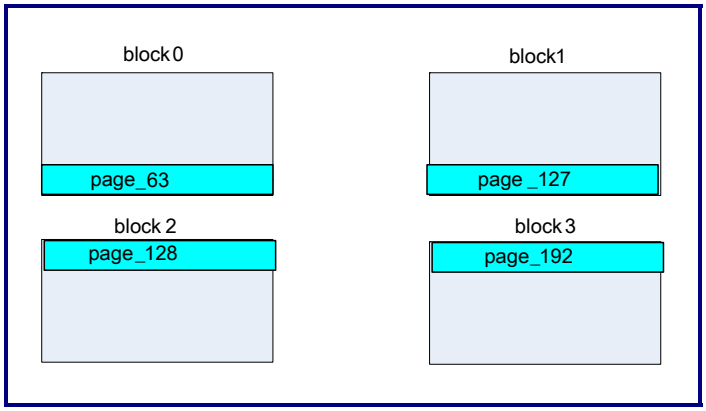



**Figure 21-23. 2-plane Write 4 Pages Cross Block**

- If the 2-plane command is increased by block (For example, two plane-block erases), cmd_scale must be set to increase two blocks.
- The row address can be increased by page or block after the goback state
- The cache page read/write must not cross the blocks.
- One flow does not include two goback states.
- Each flow must have an end state.
- The BUFFER state must be inserted between opcode and the opcode states in a programming flow.
- The programming flow cannot continuously include two busy states.
- The return state is returned to the main flow and the return state must be the last state of the subroutine.

- The goto state goes to the subroutine flow. The goto state index has at most 63, programming the subroutine must not be more than 63 indexes.
- If the interleaving flow at the auto ce sel function is enabled, inc ce val must be '1'.
- Spare SRAM is starting from the spare SRAM of ce0 when auto ce sel is enabled and the interleave 2/4 flow at auto ce sel is also enabled, for example:

  Start CE → Spare CE0

  Start CE+1 → Spare CE1

  Start CE +2 → Spare CE2

  Start CE +3 → Spare CE3
- Goto_22 is a subroutine for the ICWC flow. If users want to use this subroutine for programming the flow, there must be no other subroutine after goto_22.

### Small page fix_flow

| Coommand Register setting | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Small Page | cmd_index | bmc_ignore | byte_mode | cmd_hsk_mode | 1st_offset | 2nd_offset | counter unit | update spare |
| READ ID (byte mode) | 1B | don't care | 1 | 0 | V | V | don't care | X |
| RESET | 21 | don't care | 0 | 0 | don't care | don't care | don't care | X |
| MULTI READ STATUS | 24 | don't care | 0 | 0 | don't care | don't care | don't care | X |
| READ STATUS | 29 | don't care | 0 | 0 | don't care | don't care | times/loop | X |
| PAGE READ WITH SPARE | 2E | V | 0 | V | 0 | don't care | sector | X |
| SPARE READ | 39 | don't care | 0 | 0 | don't care | don't care | sector | X |
| BYTE READ_50 | 43 | don't care | 1 | 0 | V | don't care | don't care | X |
| BYTE READ_00 | 4C | don't care | 1 | 0 | V | don't care | don't care | X |
| BYTE READ_01 | 54 | don't care | 1 | 0 | V | don't care | don't care | X |
| PAGE WRITE WITH SPARE | 5C | V | 0 | V | 0 | don't care | sector | V |
| SPARE WRITE | 67 | don't care | 0 | 0 | don't care | don't care | sector | V |
| BYTE WRITE_50 | 71 | don't care | 1 | 0 | V | don't care | sector | X |
| BYTE WRITE_00 | 7A | don't care | 1 | 0 | V | don't care | sector | X |
| BYTE WRITE_01 | 83 | don't care | 1 | 0 | V | don't care | sector | X |
| COPY BACK | 8C | don't care | 0 | 0 | don't care | don't care | sector | X |
| HW COPY BACK | 97 | 1 | 0 | 0 | 0 | don't care | sector | V |
| BLOCK ERASE | A9 | don't care | 0 | 0 | don't care | don't care | block | X |
| BLANKING CHECK | AF | don't care | 0 | 0 | don't care | don't care | page | X |

note:

1. don't care is whatever fill in 0 or 1 is OK
2. V means the value affect behavior
3. HW COPY BACK, 1st_sec_offset must be 0.

Notes:

1. The "cmd" index is simulated with default rom-code setting in the large or small page fixed flow, if users use own rom code, the "cmd" index will be changed.
2. The small-page command index is stored at another rom code, which is different from the large rom code and only one rom code can exist. Because the size of the rom code is not large enough to store all fixed flows, when both small and large page rom codes are used, the fixed flow must be reduced to integrate one rom code, including small and large fixed flows.

### 21.14.4 Spare SRAM Organization

Figure 21-24 depicts the Spare SRAM Organization that the command queue depth is 2.

Regions 0 and 1 can be used for this Spare SRAM. The spare data is stored in Spare SRAM according to different channels, regions, and CE. Every sector has a maximum of 3-byte user data. Consequently, 3 byte * 16 = 48 byte at 8k page and sector size = 512 byte format. If the host issues a flash command in one flash, the spare data per page will be the same whether it is single plane or multi plane configuration. When the host performs the byte read/write command, its spare data will always be updated at 0 byte ~ 11 byte of every CE; that is, the top 12 bytes of the total 48 bytes. The spare SRAM is 8K in total but 6K can be used.

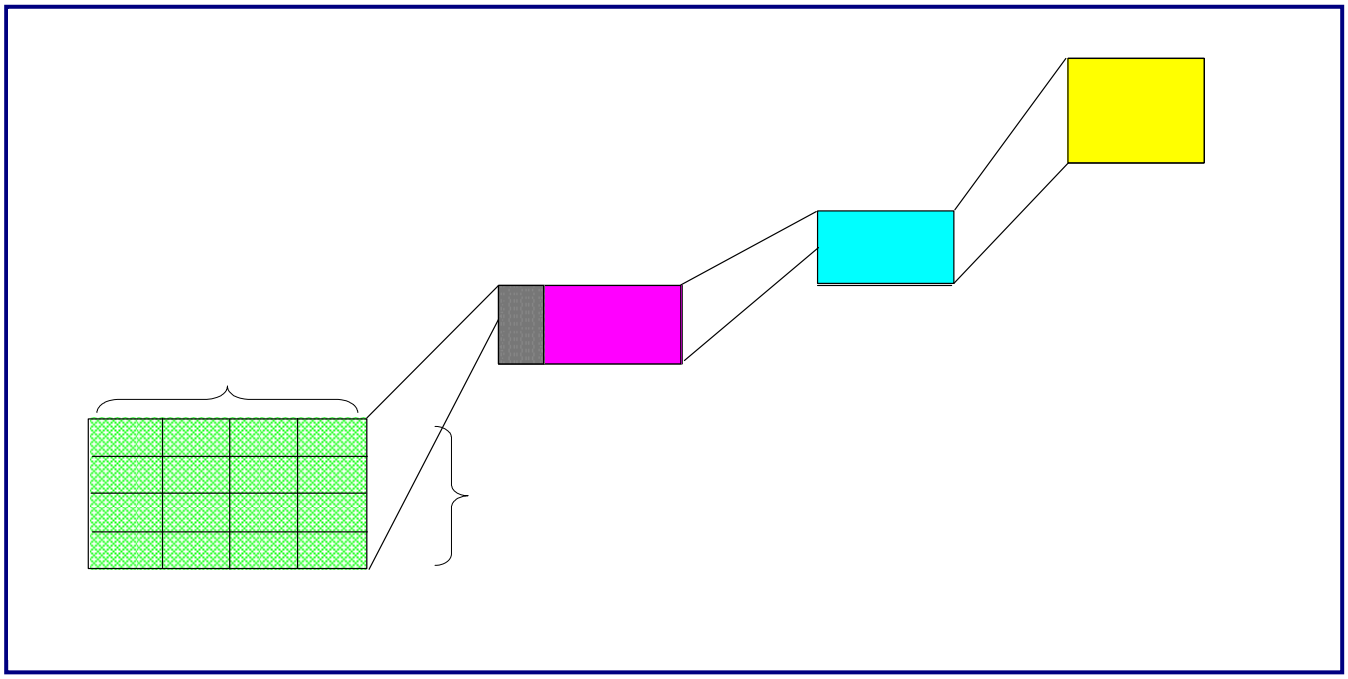


**Figure 21-24. Spare SRAM Organization of Command Queue Depth is 2**

## 21.14.5 User Data Position at Spare SRAM

### 21.14.5.1 Sector Mode

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 |

**Figure 21-25. Spare SRAM Address Mapping**

**Case a. 16 sectors**

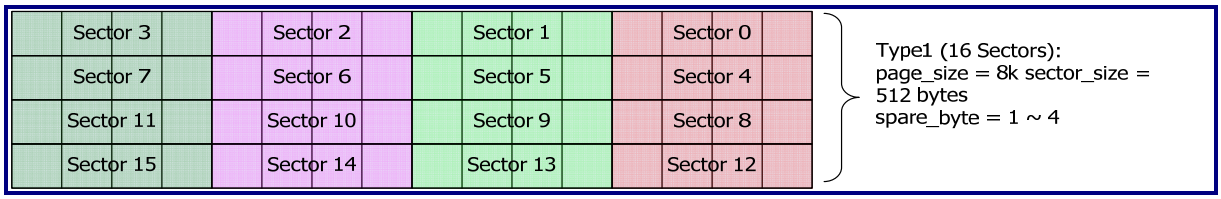



**Figure 21-26. Spare Data Position of 16 Sectors**

Numbers 0 ~ 63 are the byte index, which is up to 64 bytes. If users issue one command with the sector offset being 1 and the spare number of sectors being 3 bytes, users must fill the spare data at the correct spare SRAM position, as shown in Figure 21-27.

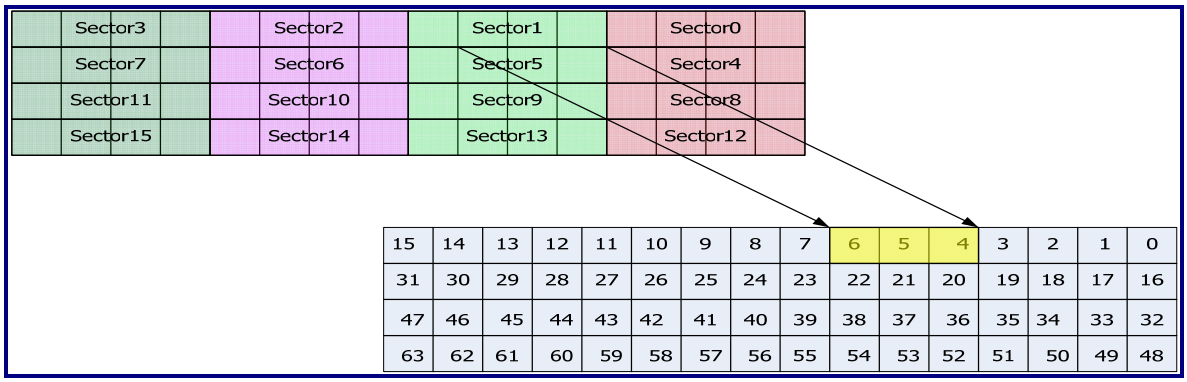



**Figure 21-27. Spare Data Position of Sector Offset 1**

Bytes 4, 5, and 6 are the spare data of sector offset 1.

### Case b. 8 sectors

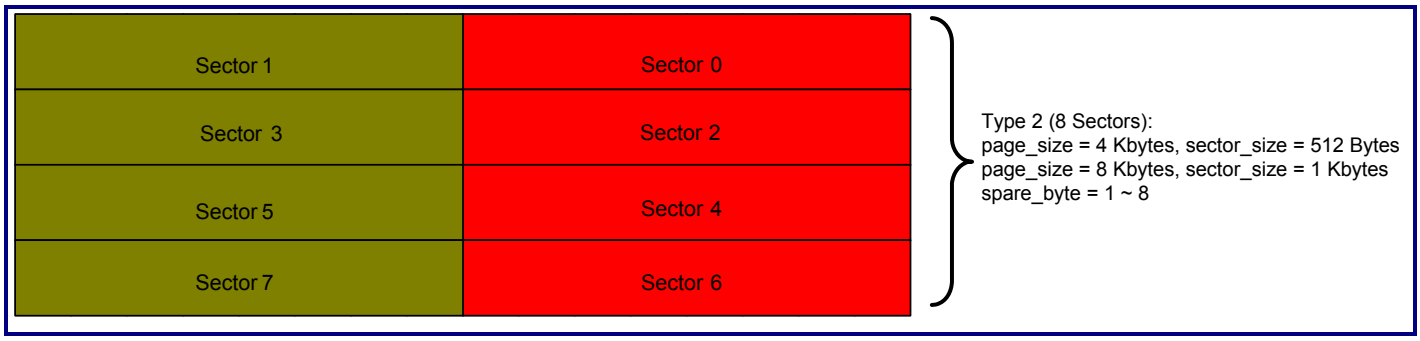



**Figure 21-28. Spare Data Position of 8 Sectors**

If users issue one command with the sector offset being 6 and if the spare number of sector is 7 bytes at 4K page and sector size is 512 bytes, users must fill the spare data at the correct spare SRAM position, as shown in Figure 21-29.

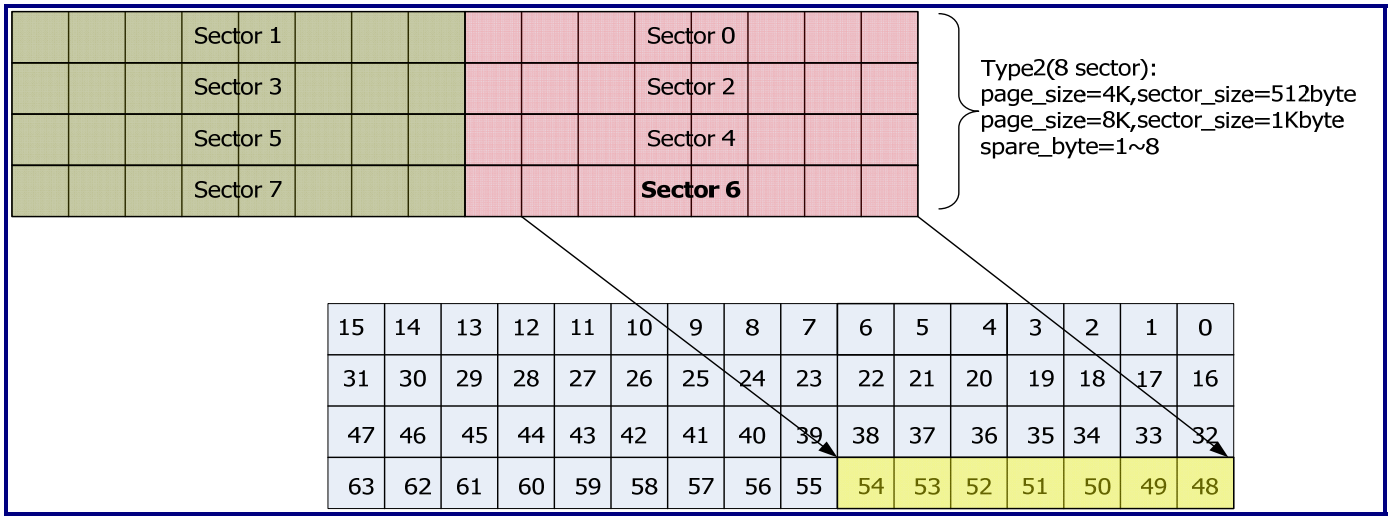



**Figure 21-29. Spare Data Position of Sector Offset 6**

Bytes 48 ~ 54 are the spare data of sector offset 6.

### Case C: 4 sectors

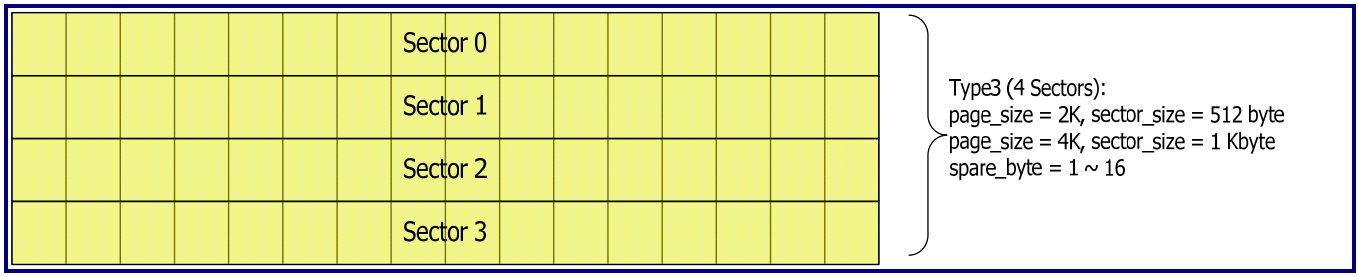



**Figure 21-30. Spare Data Position of 4 Sectors**

### Case D: 2 sectors

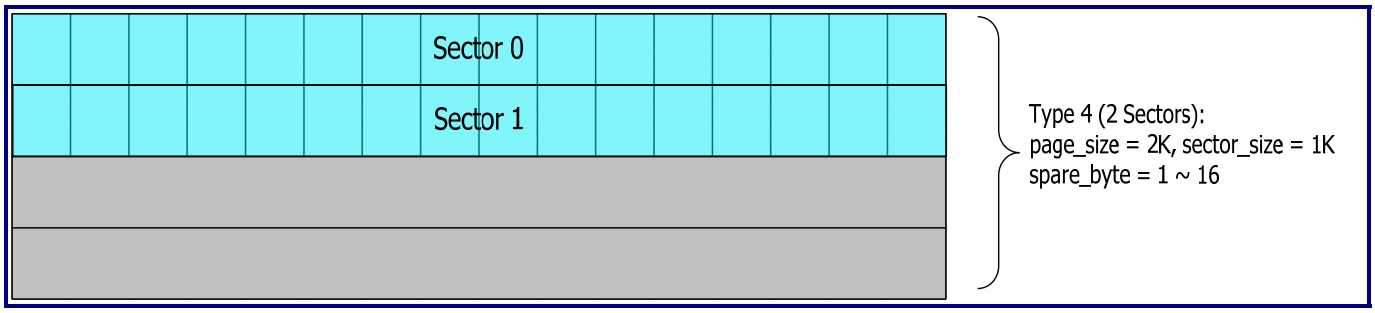



**Figure 21-31. Spare Data Position of 2 Sectors**

### Case E: 1 sector (Small page)

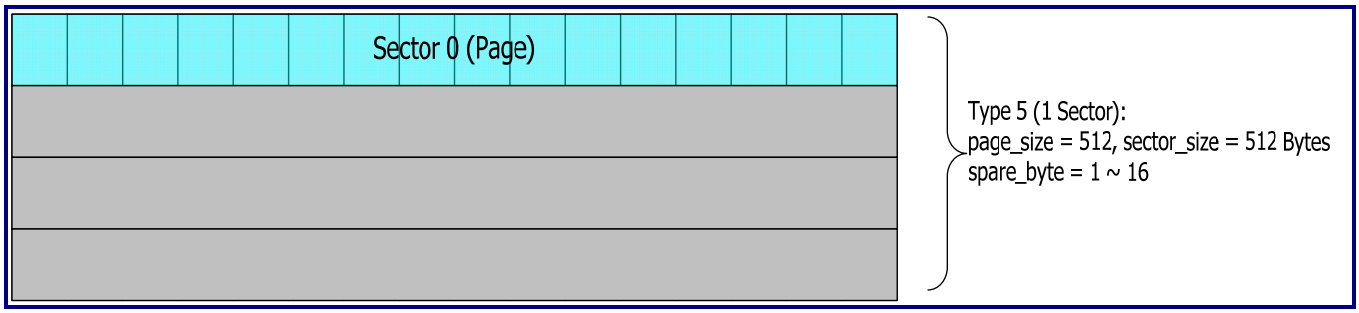



**Figure 21-32. Spare Data Position of 1 Sector**

### 21.14.5.2 Page Mode

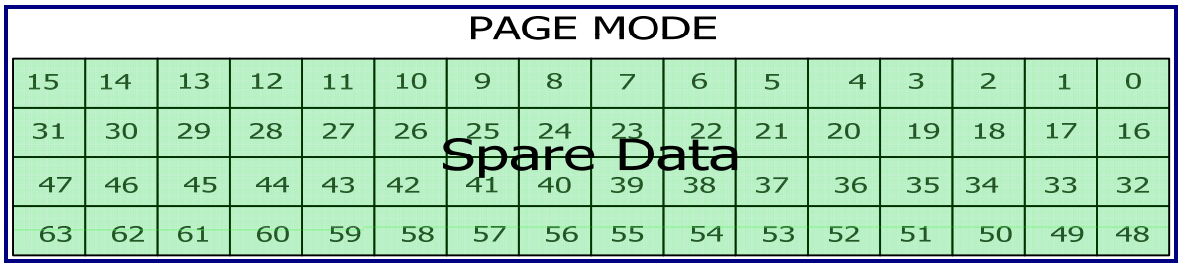



**Figure 21-33. Spare Data Position in Page Mode**

There is no sector offset in the page mode and the starting byte is at byte index 0. Users can program 4, 8, 16, 32 and 64 byte user data in the page mode. If users program 16 bytes in the page mode, the byte position will be from indexes 0 to 15.

### 21.14.6 User Data, CRC, and ECC Usages

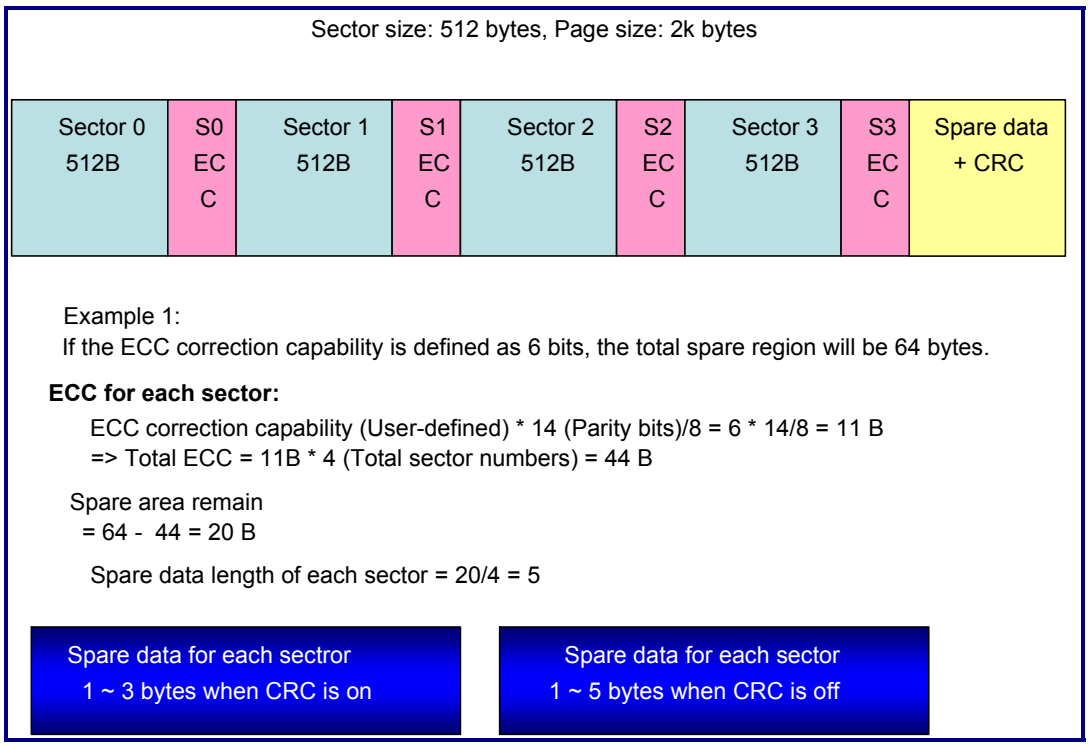



**Figure 21-34. User Data, CRC, and ECC Usages**

### 21.14.7 Fixed Flow Command and Usage

Before issuing a command, users can refer to the following fixed flow, which contains some Micro Code, and fill with the corresponding register to execute this command. For example, the first row address and second row address can act as a counter. For other settings, please refer to Table 21-60.

Basic_1

| offset | PAGE READ | PAGE WRITE WITH SPARE | SPARE WRITE | SPARE READ | PAGE READ with SPARE |
|---|---|---|---|---|---|
| 0 | OPCODE_00 | OPCODE_80 | OPOCDE_80 | OPCODE_00 | OPCODE_00 |
| 1 | 1st_col_addr(2) | 1st_col_addr(2) | sp_col_addr(2) | sp_col_addr(2) | 1st_col_addr(2) |
| 2 | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) |
| 3 | OPCODE_30 | BUFFER1 | BUFFER1 | OPCODE_30 | OPCODE_30 |
| 4 | BUSY | WDATA | WR_SP | BUSY | BUSY |
| 5 | BUFFER2 | GOTO_10(Ran In) | OPCODE_10 | BUFFER2 | BUFFER2 |
| 6 | RDATA | OPCODE_10 | BUSY | RD_SP | RDATA |
| 7 | BUFFER3 | BUSY | GOTO_0(rd_st) | BUFFER3 | BUFFER3 |
| 8 | GOBACK(8) | GOTO_0(rd_st) | GOBACK(8) | GOBACK(8) | GOTO_15(Rand_out) |
| 9 | END | GOBACK(9) | END | END | GOBACK(9) |
| 10 | | END | | | END |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |

Note: The offset number is the offset of a flow. For example, if users want to perform a PAGE READ flow and issue 1st_row_addr instead of starting from '0', only (index + offset) should be issued, which is equivalent to 0 + 2 = 2. Consequently, users can fill two command indexes in the command table.

Basic_2

| offset | READ ID (byte mode) | RESET | BLOCK ERASE | INTERNAL COPY BACK | BLANKING CHECK |
|---|---|---|---|---|---|
| 0 | OPCODE_90 | OPCODE_FF | OPCODE_60 | OPCODE_00 | OPCODE_80 |
| 1 | 1st_col_addr[00](1) | BUSY | 1st_row_addr(3) | zero_addr(2) | zero_addr(2) |
| 2 | BUFFER4 | END | OPCODE_D0 | 1st_row_addr(3) | 1st_row_addr(3) |
| 3 | RD_SP | | BUSY | OPCODE_35 | BUFFER1 |
| 4 | BUFFER3 | | GOTO_0(rd_st) | BUSY | BLANK WRITE DATA |
| 5 | END | | GOBACK(5) | OPCODE_85 | OPCODE_10 |
| 6 | | | END | sp_col_addr(2) | BUSY |
| 7 | | | | 2nd_row_addr(3) | OPCODE_00 |
| 8 | | | | BUFFER1 | zero_addr(2) |
| 9 | | | | WR_SP | 1st_row_addr(3) |
| 10 | | | | OPCODE_10 | OPCODE_30 |
| 11 | | | | BUSY | BUSY |
| 12 | | | | GOTO_0(rd_st) | BUFFER2 |
| 13 | | | | GOBACK(13) | BLANK READ DATA |
| 14 | | | | END | BUFFER3 |
| 15 | | | | | GOBACK(15) |
| 16 | | | | | END |
| 17 | | | | | |
| | | | | | NOTE: before blanking check, must do block erase |

Basic_3

| offset | byte write(byte mode) | byte read(byte mode) | multi read status | read status |
|---|---|---|---|---|
| 0 | OPCODE_80 | OPCODE_00 | GOTO_5(m_rd_st) | GOTO_0(rd_st) |
| 1 | 1st_col_addr(2) | 1st_col_addr(2) | END | GOBACK(1) |
| 2 | 1st_row_addr(3) | 1st_row_addr(3) | | END |
| 3 | BUFFER1 | OPCODE_30 | | |
| 4 | WR_SP | BUSY | | |
| 5 | OPCODE_10 | BUFFER2 | | |
| 6 | BUSY | RD_SP | | |
| 7 | GOTO_0(rd_st) | BUFFER3 | | |
| 8 | END | END | | |

2Plane_1

| offset | 2P PAGE WRITE/Samsan | 2P PAGE WRITE(Micron) | 2 PLANE ERASE/Samsan | 2 PLANE RAGE READ |
|---|---|---|---|---|
| 0 | OPCODE_80 | OPCODE_80 | OPCODE_60 | OPCODE_00 |
| 1 | 1st_col_addr(2) | 1st_col_addr(2) | fix_3rd_addr(3) | 1st_col_addr(2) |
| 2 | fix_3rd_addr(3) | 1st_row_addr(3) | OPCODE_60 | 1st_row_addr(3) |
| 3 | BUFFER1 | BUFFER1 | tog_1st_row(3) | OPCODE_30 |
| 4 | WDATA | WDATA | OPCODE_D0 | BUSY |
| 5 | GOTO_10(rand_in) | GOTO_10(rand_in) | BUSY | BUFFER2 |
| 6 | OPCODE_11 | OPCODE_11 | GOTO_0(rd_st) | RDATA |
| 7 | BUSY | BUSY | GOBACK(7) | BUFFER3 |
| 8 | OPCODE_81 | OPCODE_80 | END | GOTO_15(rand_out) |
| 9 | zero_addr(2) | zero_addr(2) | | OPCODE_00 |
| 10 | tog_1st_row(3) | 2nd_row_addr(3) | | zero_addr(2) |
| 11 | BUFFER1 | BUFFER1 | | tog_1st_row(3) |
| 12 | WDATA | WDATA | | OPCODE_30 |
| 13 | GOTO_10(rand_in) | GOTO_10(rand_in) | | BUSY |
| 14 | OPCODE_10 | OPCODE_10 | | BUFFER2 |
| 15 | BUSY | BUSY | | RDATA |
| 16 | GOTO_0(rd_st) | GOTO_0(rd_st) | | BUFFER3 |
| 17 | GOBACK(17) | GOBACK(17) | | GOTO_15(rand_out) |
| 18 | END | END | | GOBACK(18) |
| 19 | | | | END |

Please note that 2P PAGE WRITE has two different fix flows, one is for Samsung and the other is for Micron. 2P PAGE WRITE/Samsung is for Samsung and 2P PAGE WRITE/Micron is for Micron. 2P PAGE WRITE depends on the Samsung Flash specification to generate an easy and simple fixed flow. If users want to use this flow, please check if the flow matches the specification. Users should treat 2P PAGE WRITE/Micron same as the one for Samsung.

2Plane_2

| offset | 2P/I2 PAGE WRITE/Samsar | 2P COPY BACK/Samsan | 2P COPY BACK/Intel | 2P/I2 BLK ERASE/Sama | 2P/I2 BLK ERASE/Toshiba/TC |
|---|---|---|---|---|---|
| 0 | OPCODE_80 | OPCODE_60 | OPCODE_00 | OPCODE_60 | OPCODE_60 |
| 1 | 1st_col_addr(2) | fix_3rd_addr(3) | zero_addr(2) | fix_3rd_addr(3) | 1st_row_addr(3) |
| 2 | fix_3rd_row(3) | OPCODE_60 | 1st_row_addr(3) | OPCODE_60 | OPCODE_60 |
| 3 | BUFFER1 | tog_1st_row(3) | OPCODE_00 | tog_1st_row(3) | 2nd_row_addr(3) |
| 4 | WDATA | OPCODE_35 | zero_addr(2) | OPCODE_D0 | OPCODE_D0 |
| 5 | GOTO_10(rand_in) | BUSY | 2nd_row_addr(3) | INC_CE | INC_CE |
| 6 | OPCODE_11 | OPCODE_85 | OPCODE_35 | GOTO_5(m_rd_st) | GOTO_5(m_rd_st) |
| 7 | BUFFER2 | zero_addr(2) | BUSY | OPCODE_60 | OPCODE_60 |
| 8 | GOTO_5(m_rd_st) | fix_3rd_addr(3) | OPCODE_85 | fix_3rd_addr(3) | 3rd_row_addr(3) |
| 9 | OPCODE_81 | OPCODE_11 | zero_addr(2) | OPCODE_60 | OPCODE_60 |
| 10 | zero_addr(2) | BUSY | 3rd_row_addr(3) | tog_2nd_row(3) | 4th_row_addr(3) |
| 11 | tog_1st_row(3) | OPCODE_81 | OPCODE_11 | OPCODE_D0 | OPCODE_D0 |
| 12 | BUFFER1 | zero_addr(2) | BUSY | DEC_CE(ce_0) | DEC_CE(ce_0) |
| 13 | WDATA | tog_2nd_row(3) | OPCODE_85 | GOTO_5(m_rd_st) | GOTO_5(m_rd_st) |
| 14 | GOTO_10(rand_in) | OPCODE_10 | zero_addr(2) | GOBACK(14) | GOBACK(14) |
| 15 | OPCODE_10 | BUSY | 4th_row_addr(3) | INC_CE(ce_1) | INC_CE(ce_1) |
| 16 | INC_CE(ce_1) | GOTO_0(rd_st) | OPCODE_10 | GOTO_5(m_rd_st) | GOTO_5(m_rd_st) |
| 17 | GOTO_5(m_rd_st) | GOBACK(17) | BUSY | END | END |
| 18 | OPCODE_80 | END | GOTO_0(rd_st) | | |
| 19 | zero_addr(2) | | GOBACK_19 | | |
| 20 | fix_4th_row(3) | | END | | |
| 21 | BUFFER1 | | | | |
| 22 | WDATA | | | | |
| 23 | GOTO_10(rand_in) | | | | |
| 24 | OPCODE_11 | | | | |
| 25 | BUFFER2 | | | | |
| 26 | GOTO_5(m_rd_st) | | | | |
| 27 | OPCODE_81 | | | | |
| 28 | zero_addr(2) | | | | |
| 29 | tog_2nd_row(3) | | | | |
| 30 | BUFFER1 | | | | |
| 31 | WDATA | | | | |
| 32 | GOTO_10(rand_in) | | | | |
| 33 | OPCODE_10 | | | | |
| 34 | DEC_CE(ce_0) | | | | |
| 35 | GOTO_5(m_rd_st) | | | | |
| 36 | GOBACK(36) | | | | |
| 37 | INC_CE(ce_1) | | | | |
| 38 | GOTO_5(m_rd_st) | | | | |
| 39 | END | | | | |

2Plane_3

| offset | 2P PAGE WRITE/Samsang_TC | 2 PLANE ERASE_TO | 2P COPY BACK/Samsang_TC | 2P/I2 PAGE WRITE_TC |
|---|---|---|---|---|
| 0 | OPCODE_80 | OPCODE_60 | OPCODE_60 | OPCODE_80 |
| 1 | 1st_col_addr(2) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_col_addr(2) |
| 2 | 1st_row_addr(3) | OPCODE_60 | OPCODE_60 | 1st_row_addr(3) |
| 3 | BUFFER1 | 2nd_row_addr(3) | 2nd_row_addr(3) | BUFFER1 |
| 4 | WDATA | OPCODE_D0 | OPCODE_35 | WDATA |
| 5 | GOTO_10(rand_in) | BUSY | BUSY | GOTO_10(rand_in) |
| 6 | OPCODE_11 | GOTO_0(rd_st) | OPCODE_85 | OPCODE_11 |
| 7 | BUSY | GOBACK(7) | zero_addr(2) | BUFFER2 |
| 8 | OPCODE_81 | END | 3rd_row_addr(3) | GOTO_5(m_rd_st) |
| 9 | zero_addr(2) | | OPCODE_11 | OPCODE_81 |
| 10 | 2nd_row_addr(3) | | BUSY | zero_addr(2) |
| 11 | BUFFER1 | | OPCODE_81 | 2nd_row_addr(3) |
| 12 | WDATA | | zero_addr(2) | BUFFER1 |
| 13 | GOTO_10(rand_in) | | 4th_row_addr(3) | WDATA |
| 14 | OPCODE_10 | | OPCODE_10 | GOTO_10(rand_in) |
| 15 | BUSY | | BUSY | OPCODE_10 |
| 16 | GOTO_0(rd_st) | | GOTO_0(rd_st) | INC_CE(ce_1) |
| 17 | GOBACK(17) | | GOBACK(17) | GOTO_5(m_rd_st) |
| 18 | END | | END | OPCODE_80 |
| 19 | | | | zero_addr(2) |
| 20 | | | | 3rd_row_addr(3) |
| 21 | | | | BUFFER1 |
| 22 | | | | WDATA |
| 23 | | | | GOTO_10(rand_in) |
| 24 | | | | OPCODE_11 |
| 25 | | | | BUFFER2 |
| 26 | | | | GOTO_5(m_rd_st) |
| 27 | | | | OPCODE_81 |
| 28 | | | | zero_addr(2) |
| 29 | | | | 4th_row_addr(3) |
| 30 | | | | BUFFER1 |
| 31 | | | | WDATA |
| 32 | | | | GOTO_10(rand_in) |
| 33 | | | | OPCODE_10 |
| 34 | | | | DEC_CE(ce_0) |
| 35 | | | | GOTO_5(m_rd_st) |
| 36 | | | | GOBACK(36) |
| 37 | | | | INC_CE(ce_1) |
| 38 | | | | GOTO_5(m_rd_st) |
| 39 | | | | END |

Interleaving_1

| offset | I2 PAGE READ | I2 PAGE WRITE | I4 PAGE READ | I4 PAGRE WRITE |
|---|---|---|---|---|
| 0 | OPCODE_00 | OPCODE_80 | OPCODE_00 | OPCODE_80 |
| 1 | 1st_col_addr(2) | 1st_col_addr(2) | 1st_col_addr(2) | 1st_col_addr(2) |
| 2 | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) |
| 3 | OPCODE_30 | BUFFER1 | OPCODE_30 | BUFFER1 |
| 4 | INC_CE | WDATA | INC_CE | WDATA |
| 5 | OPCODE_00 | GOTO_10(rand_in) | OPCODE_00 | GOTO_10(rand_in) |
| 6 | zero_addr(2) | OPCODE_10 | zero_addr(2) | OPCODE_10 |
| 7 | 2nd_row_addr(3) | INC_CE | 2nd_row_addr(3) | INC_CE |
| 8 | OPCODE_30 | GOTO_5(m_rd_st) | OPCODE_30 | GOTO_5(m_rd_st) |
| 9 | DEC_CE | OPCODE_80 | INC_CE | OPCODE_80 |
| 10 | GOTO_5(m_rd_st) | zero_addr(2) | OPCODE_00 | zero_addr(2) |
| 11 | OPCODE_00 | 2nd_row_addr(3) | zero_addr(2) | 2nd_row_addr(3) |
| 12 | BUFFER4 | BUFFER1 | 3rd_row_addr(3) | BUFFER1 |
| 13 | RDATA | WDATA | OPCODE_30 | WDATA |
| 14 | BUFFER3 | GOTO_10(rand_in) | INC_CE | GOTO_10(rand_in) |
| 15 | GOTO_15(rand_out) | OPCODE_10 | OPCODE_00 | OPCODE_10 |
| 16 | INC_CE | DEC_CE | zero_addr(2) | INC_CE |
| 17 | GOTO_5(m_rd_st) | GOTO_5(m_rd_st) | 4th_row_addr(3) | GOTO_5(m_rd_st) |
| 18 | OPCODE_00 | GOBACK(18) | OPCODE_30 | OPCODE_80 |
| 19 | BUFFER4 | INC_CE | DEC_CE | zero_addr(2) |
| 20 | RDATA | GOTO_5(m_rd_st) | GOTO_5(m_rd_st) | 3rd_row_addr(3) |
| 21 | BUFFER3 | END | OPCODE_00 | BUFFER1 |
| 22 | GOTO_15(rand_out) | | BUFFER4 | WDATA |
| 23 | DEC_CE | | RDATA | GOTO_10(rand_in) |
| 24 | GOBACK(24) | | BUFFER3 | OPCODE_10 |
| 25 | END | | GOTO_15(rand_out) | INC_CE |
| 26 | | | INC_CE | GOTO_5(m_rd_st) |
| 27 | | | GOTO_5(m_rd_st) | OPCODE_80 |
| 28 | | | OPCODE_00 | zero_addr(2) |
| 29 | | | BUFFER4 | 4th_row_addr(3) |
| 30 | | | RDATA | BUFFER1 |
| 31 | | | BUFFER3 | WDATA |
| 32 | | | GOTO_15(rand_out) | GOTO_10(rand_in) |
| 33 | | | INC_CE | OPCODE_10 |
| 34 | | | GOTO_5(m_rd_st) | DEC_CE(ce_0) |
| 35 | | | OPCODE_00 | GOTO_5(m_rd_st) |
| 36 | | | BUFFER4 | GOBACK(36) |
| 37 | | | RDATA | INC_CE(ce_1) |
| 38 | | | BUFFER3 | GOTO_5(m_rd_st) |
| 39 | | | GOTO_15(rand_out) | INC_CE(ce_2) |
| 40 | | | INC_CE | GOTO_5(m_rd_st) |
| 41 | | | GOTO_5(m_rd_st) | INC_CE(ce_3) |
| 42 | | | OPCODE_00 | GOTO_5(m_rd_st) |
| 43 | | | BUFFER4 | END |
| 44 | | | RDATA | |
| 45 | | | BUFFER3 | |
| 46 | | | GOTO_15(rand_out) | |
| 47 | | | DEC_CE | |
| 48 | | | GOBACK(48) | |
| 49 | | | END | |

Interleaving_2

| offset | I4 BLOCK ERASE | I2 BLOCK ERASE | I2 HW COPY BACK |
|---|---|---|---|
| 0 | OPCODE_60 | OPCODE_60 | OPCODE_00 |
| 1 | 1st_row_addr(3) | 1st_row_addr(3) | zero_addr(2) |
| 2 | OPCODE_D0 | OPCODE_D0 | 1st_row_addr(3) |
| 3 | INC_CE(ce_1) | INC_CE | OPCODE_30 |
| 4 | GOTO_5(m_rd_st) | GOTO_5(m_rd_st) | BUFFER2 |
| 5 | OPCODE_60 | OCPODE_60 | GOTO_5(m_rd_st) |
| 6 | 2nd_row_addr(3) | 2nd_row_addr(3) | OPCODE_00 |
| 7 | OPCODE_D0 | OPCODE_D0 | BUFFER4 |
| 8 | INC_CE(ce_2) | DEC_CE | RDATA |
| 9 | GOTO_5(m_rd_st) | GOTO_5(m_rd_st) | BUFFER3 |
| 10 | OPCODE_60 | GOBACK(10) | OPCODE_80 |
| 11 | 3rd_row_addr(3) | INC_CE | zero_addr(2) |
| 12 | OPCODE_D0 | GOTO_5(m_rd_st) | 2nd_row_addr(3) |
| 13 | INC_CE(ce_3) | END | BUFFER1 |
| 14 | GOTO_5(m_rd_st) | | WDATA |
| 15 | OPCODE_60 | | GOTO_10(rand_in) |
| 16 | 4th_row_addr(3) | | OPCODE_10 |
| 17 | OPCODE_D0 | | INC_CE(ce_1) |
| 18 | DEC_CE(ce_0) | | GOTO_5(m_rd_st) |
| 19 | GOTO_5(m_rd_st) | | OPCODE_00 |
| 20 | GOBACK(20) | | zero_addr(2) |
| 21 | INC_CE(ce_1) | | 1st_row_addr(3) |
| 22 | GOTO_5(m_rd_st) | | OPCODE_30 |
| 23 | INC_CE(ce_2) | | BUFFER2 |
| 24 | GOTO_5(m_rd_st) | | GOTO_5(m_rd_st) |
| 25 | INC_CE(ce_3) | | OPCODE_00 |
| 26 | GOTO_5(m_rd_st) | | BUFFER4 |
| 27 | END | | RDATA |
| 28 | | | BUFFER3 |
| 29 | | | OPCODE_80 |
| 30 | | | zero_addr(2) |
| 31 | | | 2nd_row_addr(3) |
| 32 | | | BUFFER1 |
| 33 | | | WDATA |
| 34 | | | GOTO_10(rand_in) |
| 35 | | | OPCODE_10 |
| 36 | | | DEC_CE(ce_0) |
| 37 | | | GOTO_5(m_rd_st) |
| 38 | | | GOBACK(38) |
| 39 | | | INC_CE(ce_1) |
| 40 | | | GOTO_5(m_rd_st) |
| 41 | | | END |

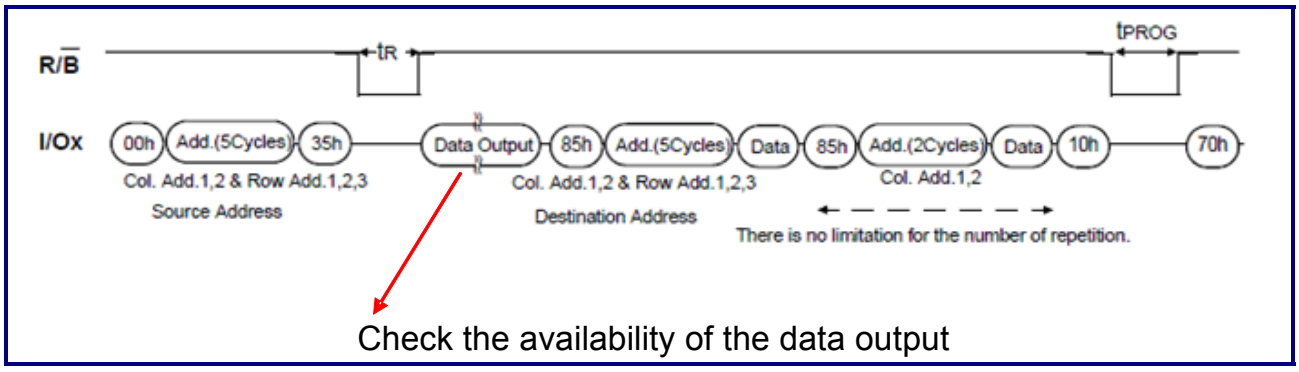
HW COPY BACK_1

| offset | HW COPY BACK | HW COPY BACK CROSS CH |
|---|---|---|
| 0 | OPCODE_00 | OPCODE_00 |
| 1 | zero_addr(2) | zero_addr(2) |
| 2 | 1st_row_addr(3) | 1st_row_addr(3) |
| 3 | OPCODE_30 | OPCODE_30 |
| 4 | BUSY | BUFFER2 |
| 5 | BUFFER2 | GOTO_5(m_rd_st) |
| 6 | RDATA | OPCODE_00 |
| 7 | BUFFER3 | BUFFER4 |
| 8 | OPCODE_80 | RDATA |
| 9 | zero_addr(2) | BUFFER3 |
| 10 | 2nd_row_addr(3) | INC_CE(ce_1) |
| 11 | BUFFER1 | GOTO_5(m_rd_st) |
| 12 | WDATA | OPCODE_80 |
| 13 | GOTO_10(rand_in) | zero_addr(2) |
| 14 | OPCODE_10 | 2nd_row_addr(3) |
| 15 | BUSY | BUFFER1 |
| 16 | GOTO_0(rd_st) | WDATA |
| 17 | GOBACk(17) | GOTO_10(rand_in) |
| 18 | END | OPCODE_10 |
| 19 | | DEC_CE(ce_0) |
| 20 | | GOBACK(20) |
| 21 | | INC_CE(ce_1) |
| 22 | | BUSY |
| 23 | | GOTO_0(rd_st) |
| 24 | | END |

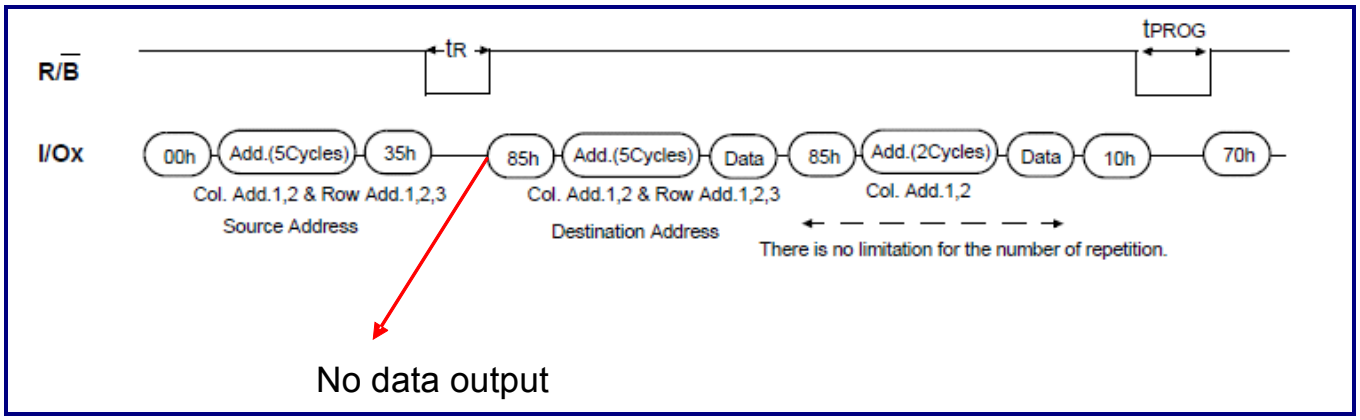
### 21.14.7.1 ICWC Fixed Flow Usage

Copy Back: opcode_00->source address->opcode_35->opcode_85->target address->opcode_10

Users must check the Flash specification for the copy-back command and make sure that after the opcode_35 data output is allowed before executing the ICWC flow. The following figure is an example.



If the opcode_35 data output is not available, the ICWC flow will not be executed. The following figure is an example.




ICWC_1

| offset | INTERNAL COPY BACK with Correct |
|---|---|
| 0 | OPCODE_00 |
| 1 | zero_addr(2) |
| 2 | 1st_row_addr(3) |
| 3 | OPCODE_35 |
| 4 | BUSY |
| 5 | BUFFER2 |
| 6 | RDATA |
| 7 | BUFFER3 |
| 8 | OPCODE_85 |
| 9 | sp_col_addr(2) |
| 10 | 2nd_row_addr(3) |
| 11 | BUFFER1 |
| 12 | WR_SP |
| 13 | GOTO_22 |
| 14 | OPCODE_10 |
| 15 | BUSY |
| 16 | OPCODE_70 |
| 17 | BUFFER4 |
| 18 | RD_ST |
| 19 | BUFFER3 |
| 20 | GOBACK(20) |
| 21 | END |

Cache_1

| offset | CACHE READ/Toshiba/Micro | CACHE WRITE/Toshiba/Micro | CAHCE COPY BACK/Toshiba/Micr |
|---|---|---|---|
| 0 | OPCODE_00 | OPCODE_80*** | OPCODE_00 |
| 1 | 1st_col_addr(2) | 1st_col_addr(2) | 1st_col_addr(2) |
| 2 | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) |
| 3 | OPCODE_30 | BUFFER1 | OPCODE_30 |
| 4 | BUSY | WDATA | BUSY |
| 5 | BUFFER2 | GOTO_10(rand_in) | BUFFER2 |
| 6 | OPCODE_31*** | OPCODE_15 | OPCODE_8C |
| 7 | BUSY | BUSY | zero_addr(2) |
| 8 | BUFFER2 | GOTO_0(rd_st) | 2nd_row_addr(3) |
| 9 | RDATA | GOBACK(9) | BUFFER1 |
| 10 | BUFFER3 | RELOAD | OPCODE_15 |
| 11 | GOTO_15(rand_out) | OPCODE_80 | BUSY |
| 12 | GOBACK(6) | zero_addr(2) | OPCODE_00 |
| 13 | RELOAD | 1st_row_addr(3) | zero_addr(2) |
| 14 | OPCODE_3F | BUFFER1 | 3rd_row_addr(3)[1st_row_addr+ |
| 15 | BUSY | WDATA | OPCODE_3A |
| 16 | BUFFER2 | GOTO_10(rand_in) | BUSY |
| 17 | RDATA | OPCODE_10 | BUFFER3 |
| 18 | BUFFER3 | BUSY | OPCODE_8C |
| 19 | GOTO_15(rand_out) | GOTO_0(rd_st) | zero_addr(2) |
| 20 | END | END | 4th_row_addr(3)[2nd_row_addr |
| 21 | | | BUFFER1 |
| 22 | | | OPCODE_15 |
| 23 | | | BUSY |
| 24 | | | GOBACK(12) |
| 25 | | | RELOAD |
| 26 | | | OPCODE_00 |
| 27 | | | zero_addr(2) |
| 28 | | | 3rd_row_addr(3) |
| 29 | | | OCPODE_3A |
| 30 | | | BUSY |
| 31 | | | BUFFER2 |
| 32 | | | OPCODE_8C |
| 33 | | | zero_addr(2) |
| 34 | | | 4th_row_addr(3) |
| 35 | | | BUFFER1 |
| 36 | | | OPCODE_10 |
| 37 | | | BUSY |
| 38 | | | GOTO_0(rd_st) |
| 39 | | | END |

ONFI_1

| offset | SYNCHRONOUS RESI | READ PARAMETER PA( | READ UNIQUE I | GET FEATURES | SET FEATURES | SELECT LUN WITH STATU |
|---|---|---|---|---|---|---|
| 0 | OPCODE_FC | OPCODE_EC | OPCODE_ED | OPCODE_EE | OPCODE_EF | OPCODE_78 |
| 1 | BUSY | 1st_row_addr(1) | 1st_row_addr(1 | 1st_row_addr(1) | 1st_row_addr(1) | 1st_row_addr(3) |
| 2 | END | BUSY | BUSY | BUSY | BUFFER1 | BUFFER4 |
| 3 | | BUFFER2 | BUFFER2 | BUFFER2 | WR_SP | RD_ST |
| 4 | | RDATA | RDATA | RD_SP | BUSY | BUFFER3 |
| 5 | | BUFFER3 | BUFFER3 | BUFFER3 | END | END |
| 6 | | END | END | END | | |

Note : Users must set no_ecc_parity and sector counter according to the specification when executing READ_PARAMSTER_PAGE/READ UNIQUE ID.

**Subroutine Command Index**

| offset | read status |
|---|---|
| 0 | opcode_70 |
| 1 | buffer4(tWHR) |
| 2 | rd_st |
| 3 | buffer3(tRHZ) |
| 4 | RETURN |
| offset | multi read status |
| 5 | opcode 70 |
| 6 | buffer4(tWHR) |
| 7 | m_rd_st |
| 8 | buffer3(tRHZ) |
| 9 | RETURN |
| offset | random data in |
| 10 | OPCODE 85 |
| 11 | sp_col_addr(2) |
| 12 | BUFFER1 |
| 13 | WR_SP |
| 14 | RETURN |
| offset | random data out |
| 15 | OPCODE_05 |
| 16 | sp_col_addr(2) |
| 17 | OPCODE_E0 |
| 18 | BUFFER4 |
| 19 | RD_SP |
| 20 | BUFFER3 |
| 21 | RETURN |
| offset | icwc |
| 22 | OPCODE_85 |
| 23 | 2nd_col_addr(2) |
| 24 | BUFFER1 |
| 25 | WDATA |
| 26 | RETURN |

| Sub-Routine | cmd_index |
|---|---|
| Read Status | 0 |
| Multi Read Status | 5 |
| Random Data In | 10 |
| Random Data Output | 15 |
| ICWC | 22 |

Subroutine Usage: When the main flow executes goto_5, it will issue multiple read status subroutines and finally return to the main flow.

**Small Page Flow**

Basic

| offset | READ ID(byte mod | RESET | Multi Read Status | OPCODE_70 |
|---|---|---|---|---|
| 0 | OPCODE_90 | OPCODE_FF | OPCODE_70 | BUFFER_4 |
| 1 | COL_1ST | BUSY | BUFFER_4 | RD_ST |
| 2 | BUFFER_4 | END | M_RD_ST | BUFFER_3 |
| 3 | RD_SP | | BUFFER_3 | END |
| 4 | BUFFER_3 | | END | |
| 5 | END | | | |

Page read

| offset | PAGE READ | spare_read | byte_read(byte_mode | byte_read(byte_mode | byte read(byte mode) |
|---|---|---|---|---|---|
| 0 | OPCODE_00 | OPCODE_50 | OPCODE_50 | OPCODE_00 | OPCODE_01 |
| 1 | 1st_col_addr(1) | sp_addr(1) | 1st_col_addr(1) | 1st_col_addr(1) | 1st_col_addr(1) |
| 2 | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) |
| 3 | BUSY | BUSY | BUSY | BUSY | BUSY |
| 4 | BUFFER | BUFFER | BUFFER | BUFFER | BUFFER |
| 5 | RDATA | RD_SP | RD_SP | RD_SP | RD_SP |
| 6 | RD_SP | BUFFER | BUFFER(tRHZ) | BUFFER(tRHZ) | BUFFER(tRHZ) |
| 7 | BUFFER | BUSY | BUSY | END | END |
| 8 | BUSY | GOBACK(8) | END | | |
| 9 | GOBACK(9) | END | | | |
| 10 | END | | | | |

Page write

| offset | PAGE WRITE WITH SPAR | spare_write | byte_write(byte_mode | byte_write(byte_mode | byte_write(byte_mode |
|---|---|---|---|---|---|
| 0 | OPCODE_00 | OPCODE_50 | OPCODE_50 | OPCODE_00 | OPCODE_01 |
| 1 | OPCODE_80 | OPCODE_80 | OPOCDE_80 | OPOCDE_80 | OPOCDE_80 |
| 2 | 1st_col_addr(1) | sp_addr(1) | 1st_col_addr(1) | 1st_col_addr(1) | 1st_col_addr(1) |
| 3 | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) | 1st_row_addr(3) |
| 4 | BUFFER1(tADL) | BUFFER1(tADL) | BUFFER1(tADL) | BUFFER1(tADL) | BUFFER1(tADL) |
| 5 | WDATA | WR_SP | WR_SP | WR_SP | WR_SP |
| 6 | WR_SP | OPCODE_10 | OPCODE_10 | OPCODE_10 | OPCODE_10 |
| 7 | OPCODE_10 | BUSY | BUSY | BUSY | BUSY |
| 8 | BUSY | GOBACK(7) | END | END | END |
| 9 | GOBACK(8) | END | | | |
| 10 | END | | | | |

MISC_1

| offset | copy back | hw copy back | block erase | blanking check |
|---|---|---|---|---|
| 0 | OPCODE_00 | OPCODE_00 | OPCODE_60 | OPCODE_00 |
| 1 | 1st_col_addr(1) | 1st_col_addr(1) | 1st_row_addr(2) | OPCODE_80 |
| 2 | 1st_row_addr(2) | 1st_row_addr(2) | OPCODE_D0 | zero_addr(1) |
| 3 | BUSY | BUSY | BUSY | 1st_row_addr(2) |
| 4 | BUFFER2 | BUFFER2 | GOBACK(4) | BUFFER1(tADL) |
| 5 | OPCODE_8A | RDATA | END | BLANK WRITE DATA |
| 6 | 2nd_col_addr(1) | BUFFER3 | | OPCODE_10 |
| 7 | 2nd_row_addr(2) | BUSY | | BUSY |
| 8 | BUSY | OPCODE_80 | | OPCODE_00 |
| 9 | GOBACK(9) | 2nd_col_addr(1) | | zero_addr(1) |
| 10 | END | 2nd_row_addr(2) | | 1st_row_addr(2) |
| 11 | | BUFFER1(tADL) | | BUSY |
| 12 | | WDATA | | BUFFER2(tAR) |
| 13 | | WR_SP | | BLANK READ DATA |
| 14 | | OPCODE_10 | | BUFFER3(tRHZ) |
| 15 | | BUSY | | BUSY |
| 16 | | GOBACK(16) | | GOBACK(16) |
| 17 | | END | | END |

# Chapter 22

# Scaler

This chapter contains the following sections:


- 22.1 General Description
- 22.2 Features
- 22.3 Block Diagram
- 22.4 Modes of Operation
- 22.5 Programming Model
- 22.6 Function Descriptions

## 22.1 General Description

Scaler (SCAR) is in charge of resizing and enhancing video data from ITU-R BT. 656 or memory and output data to AMBA AHB. With the resizing ability, users can size-down or size-up an image to the necessary resolution or do aspect ratio conversion. The color OSD function can help users to paste any character at the captured video. The window clipping function can clip the interest region from the image before or after sizing down. Moreover, the luminance compensation, sharpness, and color saturation control can help make the image more pleasing to the human eyes.

## 22.2 Features

- Supports image resolution of up to 4096x4096
- ITU-R BT. 656 8-bit and 10-bit input interfaces
- More than 1x size up with independent fractional horizontal and vertical ratio
- Up to 1/127x size down with independent fractional horizontal and vertical ratio
- Supports non-linear resizing for aspect ratio conversion from 4:3 to 16:9
- Programming luminance compensation curve
- Supports sharpness control
- Supports Saturation and Hue control
- Supports Contrast and Brightness control
- Programming horizontal and vertical crop at source image and output image
- Supports color OSD
- Supports color space conversion from YCbCr to RGB
- Output format:
  - RGB 888/RGB 565
  - YCbCr4:4:4/YCbCr4:2:2/YCbCr4:2:0
- Input format:
  - YCbCr 4:4:4/YCbCr 4:2:2
- Raster and macro block order output sequence

## 22.3 Block Diagram

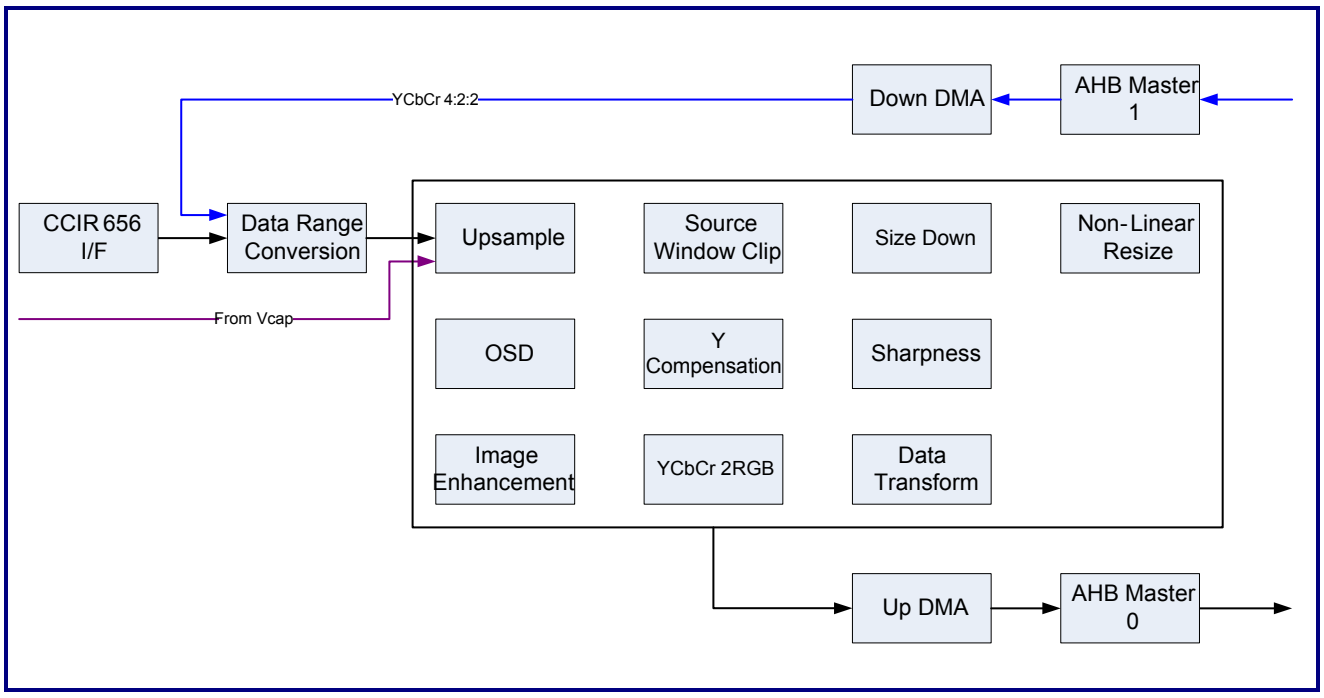



**Figure 22-1. Block Diagram**

## 22.4 Modes of Operation

### Video Capture Path

Image data are captured from the video port and the output to other AHB slave.

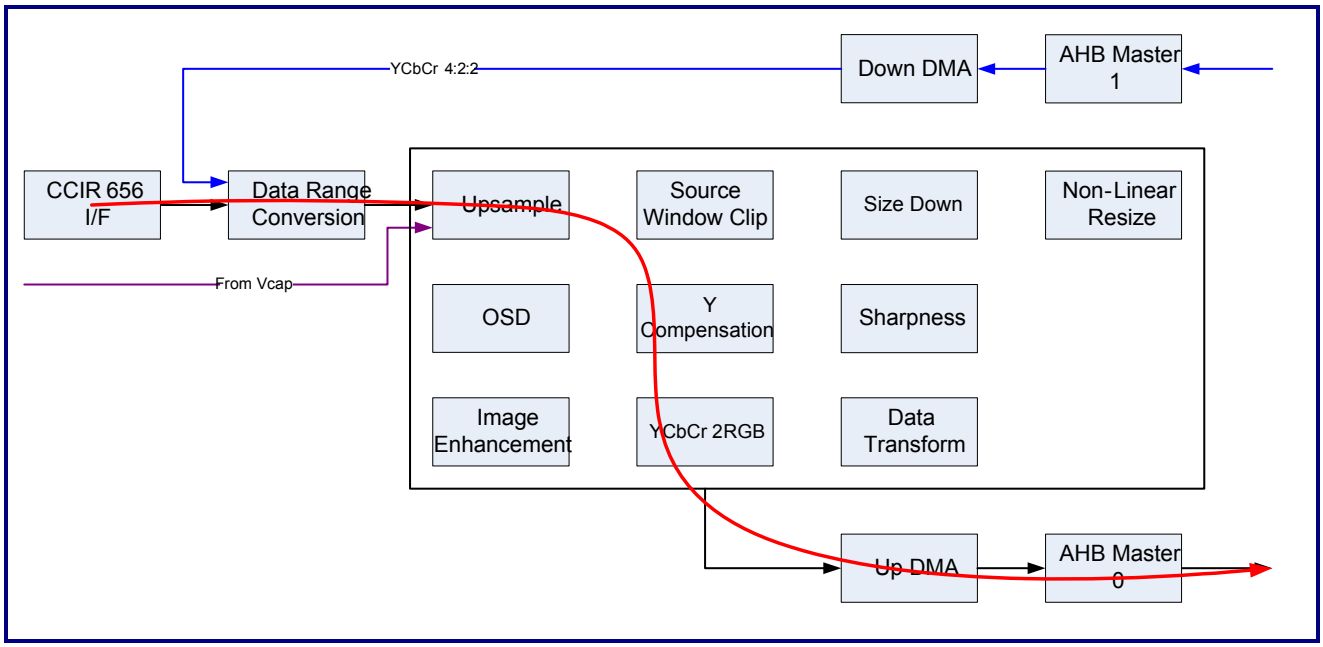



**Figure 22-2. Video Capture Path**

### Loopback Path

Image can be read through the AHB Master 1 from other AHB slave instead of from the video port. Users have to trigger SCAR by the configuration register to capture images for every frame.

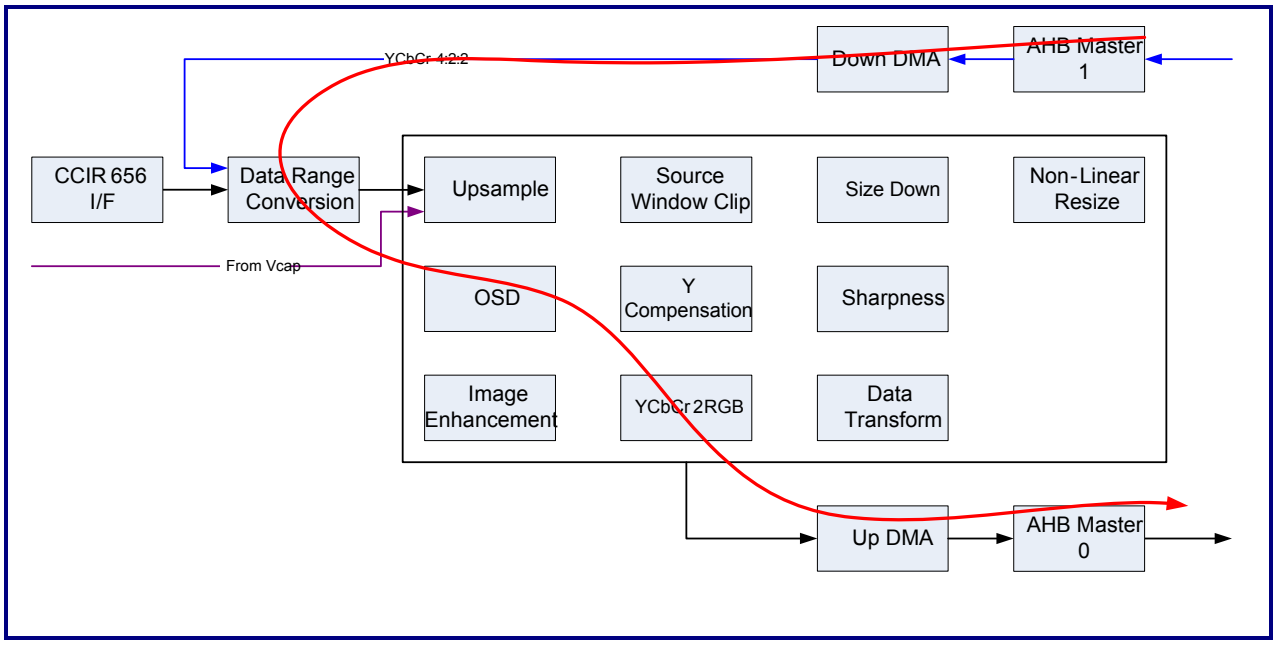



**Figure 22-3. Loopback Path**

## 22.5 Programming Model

### 22.5.1 Mapping of Usual Special-purpose Registers

**Table 22-1. Mapping of Usual Special-purpose Registers**

| Symbol | Offset | Access | Description |
|---|---|---|---|
| SCAREN | 0x0000 | RW | Basic function enable and trigger control |
| SIZE0 | 0x0020 | RW | Output target image width and height |
| SIZE1 | 0x0024 | RW | Top-left corner of output crop window |
| SIZE2 | 0x0028 | RW | Output crop width and height |
| DFORMAT | 0x0038 | RW | Data output format |
| SRCSEL | 0x003C | RW | Video source selection |
| FMRATE | 0x0040 | RW | Frame skip or block control |
| SRCIF | 0x0044 | RW | Source interface related control |
| SRCSIZE0 | 0x0050 | RW | Source image width and height |
| SRCSIZE1 | 0x0064 | RW | Top-left corner of source active window |
| SRCSIZE2 | 0x0068 | RW | Source active window width and height |
| IEGAIN | 0x0078 | RW | Y/Cb/Cr gain |
| IECTRL | 0x007c | RW | Mono control |
| YCPT0 | 0x00a0 | RW | Start point of YC map curve |
| YCPT1 | 0x00a4 | RW | Start point of YC map curve |
| YCPT2 | 0x00a8 | RW | Start point of YC map curve |
| YCSLP0 | 0x00ac | RW | Slope YC map curve |
| YCSLP1 | 0x00b0 | RW | Slope YC map curve |
| YCSLP2 | 0x00b4 | RW | Slope YC map curve |
| YCSLP3 | 0x00b8 | RW | Slope YC map curve |
| YCSLP4 | 0x00bc | RW | Slope YC map curve |
| YCSLP5 | 0x00c0 | RW | Slope YC map curve |
| YLEVEL | 0x00c4 | RW | Y level adjustment |
| MXCOE0 | 0x00c8 | RW | Color matrix coefficient of image enhancement |
| MXCOE1 | 0x00cc | RW | Color matrix coefficient of image enhancement |
| SPTH | 0x00d0 | RW | Sharpness threshold and gain |
| NLRSDIF | 0x00dc | RW | Non-linear resizing control |
| DDMA0 | 0x0200 | RW | Down stream DMA length control |
| UDMA0 | 0x0204 | RW | Up stream DMA length and sequence control |

| Symbol | Offset | Access | Description |
|---|---|---|---|
| MEMSRC0 | 0x0208 | RW | Memory source format control for loop-back path |
| MEMSRC1 | 0x020C | RW | Memory source image start address for loop-back path |
| MEMSRC2 | 0x0214 | RW | Memory source image size for loop-back path |
| DMA1 | 0x021C | RW | Pitch of destination window |
| MDEST0 | 0x0250 | RW | Destination frame buffer 0 Y start address |
| MDEST1 | 0x0254 | RW | Destination frame buffer 1 Y start address |
| MDEST2 | 0x0260 | RW | Destination frame buffer 0 Cb start address |
| MDEST3 | 0x0264 | RW | Destination frame buffer 1 Cb start address |
| MDEST4 | 0x0270 | RW | Destination frame buffer 0 Cr start address |
| MDEST5 | 0x0274 | RW | Destination frame buffer 1 Cr start address |
| OSDFONT | 0x0300 | RW | OSD font RAM address and data port |
| OSDDISP | 0x0304 | RW | OSD display RAM address and data port |
| OSDREAD | 0x0308 | R | OSD font RAM and display RAM data read port |
| OSDEN | 0x030C | RW | OSD window 0 ~ window 3 enable |
| OSDPAT0 | 0x0310 | RW | OSD palette color 0 |
| OSDPAT1 | 0x0314 | RW | OSD palette color 1 |
| OSDPAT2 | 0x0318 | RW | OSD palette color 2 |
| OSDPAT3 | 0x031C | RW | OSD palette color 3 |
| OSDPAT4 | 0x0320 | RW | OSD palette color 4 |
| OSDPAT5 | 0x0324 | RW | OSD palette color 5 |
| OSDPAT6 | 0x0328 | RW | OSD palette color 6 |
| OSDCOR0 | 0x0330 | RW | OSD font color of window 0 |
| OSDWSZ0 | 0x0334 | RW | OSD window 0 width/height |
| OSDSSZ0 | 0x0338 | RW | OSD window 0 start point |
| OSDFSZ0 | 0x033C | RW | OSD font size of window 0 |
| OSDCOL1 | 0x0340 | RW | OSD font color of window 1 |
| OSDWSZ1 | 0x0344 | RW | OSD window 1 width/height |
| OSDSSZ1 | 0x0348 | RW | OSD window 1 start point |
| OSDFSZ1 | 0x034C | RW | OSD font size of window 1 |
| OSDCOR2 | 0x0350 | RW | OSD font color of window 2 |
| OSDWSZ2 | 0x0354 | RW | OSD window 2 width/height |
| OSDSSZ2 | 0x0358 | RW | OSD window 2 start point |
| OSDFSZ2 | 0x035C | RW | OSD font size of window 2 |
| OSDCOR3 | 0x0360 | RW | OSD font color of window 3 |

| Symbol | Offset | Access | Description |
|---|---|---|---|
| OSDWSZ3 | 0x0364 | RW | OSD window 3 width/height |
| OSDSSZ3 | 0x0368 | RW | OSD window 3 start point |
| OSDFSZ3 | 0x036C | RW | OSD font size of window 3 |
| INTSTS | 0x03E0 | RW | Interrupt status register |
| INTMASK | 0x03E4 | RW | Interrupt mask register |
| SCARSTS | 0x03EC | RW | Scaler internal status |

## 22.5.2 Special-purpose Registers and Functions

### 22.5.2.1 SCAREN (Offset: 0x0000)

**Table 22-2. SCAREN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | SCAR_EN | R/W | 0 | Enable Scaler<br>0: Disable the Scaler<br>1: Enable the Scaler |
| [30:29] | - | - | - | Reserved |
| 28 | ALBUM_EN | R/W | 0 | Photo album capture mode enable (Sync. register)<br>0: Disable the photo album mode<br>1: Enable the photo album mode |
| [27:25] | - | - | - | Reserved |
| 24 | PIMG_EN | R/W | 0 | Post-image processing fire (Auto clear)<br>1: Fire post-image processing |
| [23:13] | - | - | - | Reserved |
| 12 | VP_EC_EN | R/W | 1 | Enable the error correction of the CCIR 656 SAV/EAV code<br>0: Disable the error correction<br>1: Enable the error correction |
| [11:10] | - | - | - | Reserved |
| 9 | US_EN | R/W | 1 | Enable CbCr up-sampling algorithm<br>0: Resample CbCr by duplicated method<br>1: Resample CbCr by up-sampling algorithm |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 8 | DITHER_EN | R/W | 1 | Enable dither function for DT module<br>0: Disable the dither<br>1: Enable the dither |
| 7 | - | - | - | Reserved |
| 6 | SP_EN | R/W | 1 | Enable sharpness function (Sync. register)<br>0: Disable sharpness<br>1: Enable sharpness |
| 5 | - | - | - | Reserved |
| 4 | NLRS_EN | R/W | 0 | Enable Non-linear resizing function (Sync. register)<br>0: Disable non-linear resizing<br>1: Enable non-linear resizing |
| [3:0] | - | - | - | Reserved |

### 22.5.2.2 SCARUPD (Offset: 0x0004)

**Table 22-3. SCARUPD Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:17] | - | - | - | Reserved |
| 16 | FORCE_CU | R/W | 0 | Force CU to perform calculation (Auto clear)<br>0: CU does calculation in the vertical blank interval<br>1: Force CU to do calculation |
| [15:8] | - | - | - | Reserved |
| [7:2] | - | - | - | Reserved |
| 1 | FORCE_UPDATE | R/W | 0 | Force all of the sync register to do update (Auto clear)<br>0: Do not update sync. register<br>1: All of the sync. register will be updated immediately |
| 0 | SYNC_REG_UPDATE | R/W | 0 | Trigger all of the sync. register (Auto clear)<br>0: Sync. register retains original value<br>1: All of sync. register will be updated at time of the frame end |

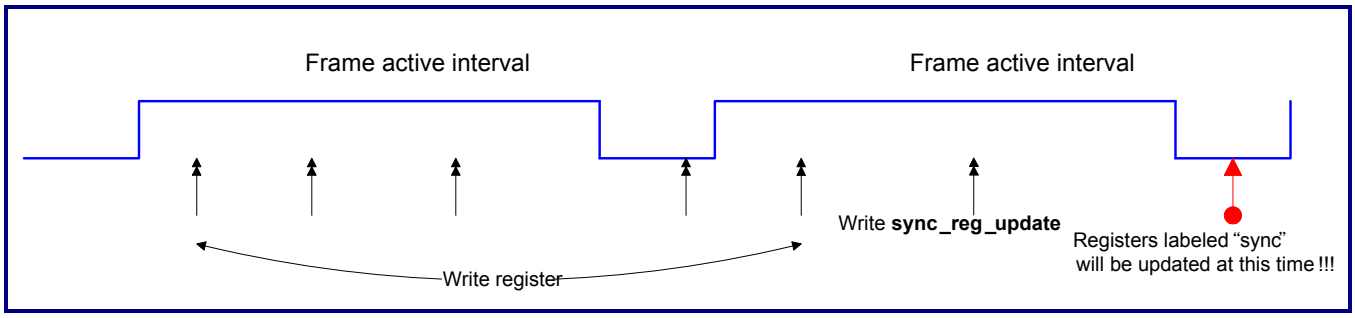



**Figure 22-4. Timing of sync_reg_update**

### 22.5.2.3 SCARCLK (Offset: 0x0008)

**Table 22-4. SCARCLK Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| 28 | SW_RST | R/W | 0 | Software reset (Auto clear)<br>1: Set 1 to reset |
| [27:24] | - | - | - | Reserved |
| [23:17] | - | - | - | Reserved |
| 16 | INT_SYNC_CLR | R/W | 0 | Clear interrupt source at every vertical start automatically<br>0: User clear<br>1: Clear automatically |
| [15:8] | - | - | - | Reserved |
| [7:5] | - | - | - | - |
| 4 | VP_CLK_INV | R/W | 0 | Invert the input video pixel clock<br>0: Disable<br>1: Enable |
| [3:0] | - | - | - | Reserved |

### 22.5.2.4 ASPMJ (Offset: 0x001C)

**Table 22-5. ASPMJ Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| [15:10] | - | - | - | Reserved |
| [9:8] | ASPECT_MAJOR | R/W | 0 | Aspect ratio control for image resizing<br>00: Resize of output image in the X-, Y-direction independent mode<br>01: Resize of output image keep aspect ratio by X-direction major<br>10: Resize of output image keep aspect ratio by X-direction major |
| [7:0] | - | - | - | Reserved |

### 22.5.2.5 SIZE0 (Offset 0x0020)

**Table 22-6. SIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | TARGET_HEIGHT | R/W | 0 | Output target image height (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:0] | TARGET_WIDTH | R/W | 0 | Output target image width (Sync. register) |

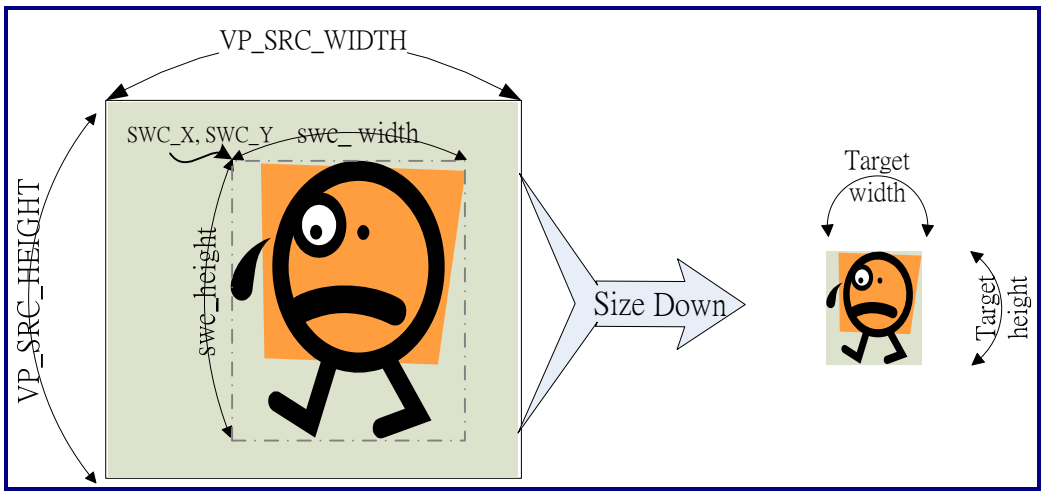



**Figure 22-5. Relationship between Source Image and Target Image**

### 22.5.2.6 SIZE1 (Offset: 0x0024)

**Table 22-7. SIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | CROP_Y | R/W | 0 | Y-coordinate of the top-left corner for the output crop window |
| [15:13] | - | - | - | Reserved |
| [12:0] | CROP_X | R/W | 0 | X-coordinate of the top-left corner for the output crop window |

### 22.5.2.7 SIZE2 (Offset: 0x0028)

**Table 22-8. SIZE2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | CROP_HEIGHT | R/W | 0 | Output crop image height in the multiple of 2 |
| [15:13] | - | - | - | Reserved |
| [12:0] | CROP_WIDTH | R/W | 0 | Output crop image width in the multiple of 4 |

### 22.5.2.8 BORDER_SIZE (Offset: 0x0030)

**Table 22-9. BORDER_SIZE Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| [15:12] | RC_BD_HEIGHT | R/W | 0 | Border height<br>00: No border at the top and bottom sides<br>001 ~ 111: Border height = (Reg value x 2) lines at both the top and bottom sides |
| [11:8] | RC_BD_WIDTH | R/W | 0 | Border width<br>00: No border at the left and right sides<br>001 ~ 111: Border width = (Reg value x 2) pixels at both the left and right sides |
| [7:0] | - | - | - | Reserved |

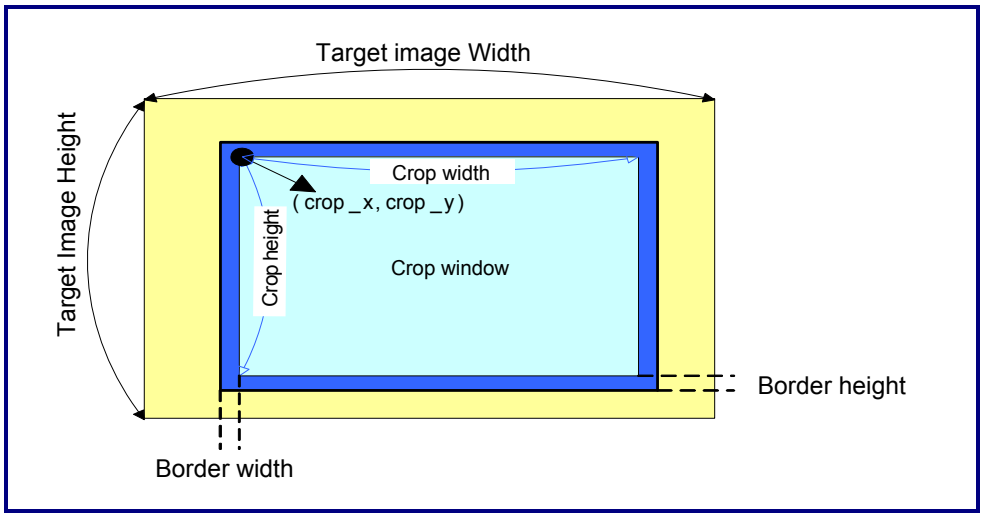



**Figure 22-6. Relationship between Border Width/Height and Crop Window**

### 22.5.2.9 BORDER_COLOR (Offset: 0x0034)

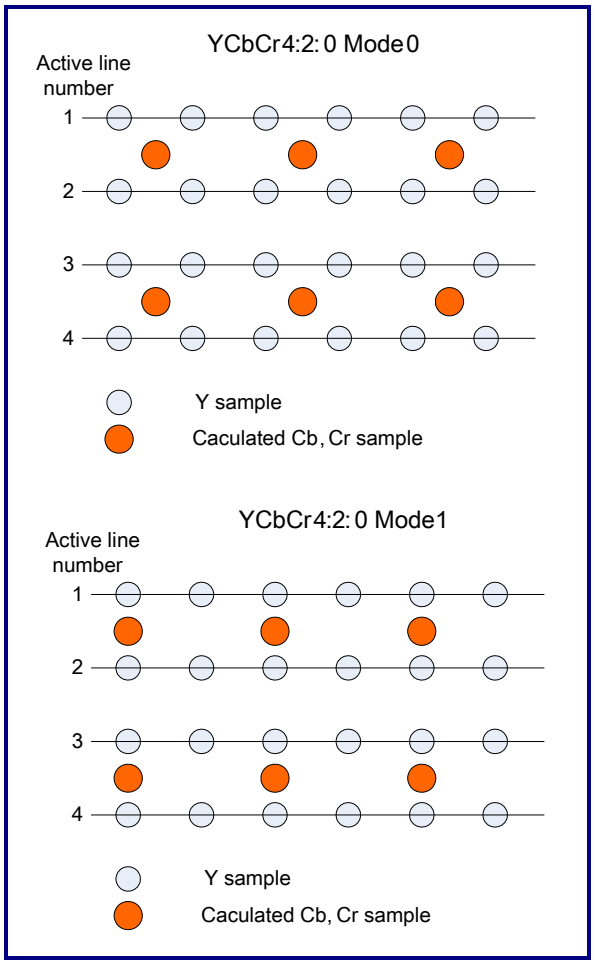
**Table 22-10. BORDER_COLOR Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | RC_BD_R_COLOR | R/W | 0 | R/Cr bits[7:3] border color |
| [26:21] | RC_BD_G_COLOR | R/W | 0 | G/Y bits[7:2] border color |
| [20:16] | RC_BD_B_COLOR | R/W | 0 | B/Cb bits[7:3] border color |
| [15:0] | - | - | - | Reserved |

### 22.5.2.10 DFORMAT (Offset 0x0038)

**Table 22-11. DFORMAT Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:16] | - | - | - | Reserved |
| [15:14] | - | - | - | Reserved |
| 13 | DEST_YC_SWAP | R/W | 0 | Destination CbCr/Y or BR/G sequence swap<br>This bit is ineffective if DMA write out YCbCr into two or three separated memory blocks.<br>0: Sequence of CbCr is ahead of Y (Cb Y Cr or Cr Y Cb)<br>1: Sequence of Y is ahead of CbCr (Y Cb Cr or Y Cr Cb) |
| 12 | DEST_RB_SWAP | R/W | 0 | Destination R/B or Cr/Cb sequences swap<br>This bit is ineffective if DMA write YCbCr data into two or three separated memory blocks.<br>0: Sequence of Cr is ahead of Cb → Cr Cb<br>1: Sequence of Cb is ahead of Cr → Cb Cr |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [11:8] | DEST_FORMAT | R/W | 0 | Data output format<br>x000: RGB 888<br>x001: RGB 565<br>x010 ~ x011: Reserved<br>x100: YCbCr 4:4:4<br>x101: YCbCr 4:2:2<br>x110: YCbCr 4:2:0 mode 0<br>x111: YCbCr 4:2:0 mode 1<br>0xxx: YCbCr 256 level<br>1xxx: YCbCr 240 level<br>  |
| [7:0] | - | - | - | Reserved |

### 22.5.2.11 SRCSEL (Offset: 0x003C)

**Table 22-12. SRCSEL Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | RS_H_BLANK_PIX | R/W | 0x08 | Horizontal blank width for the extended size-up line |
| [23:16] | - | - | - | Reserved |
| [15:10] | - | - | - | Reserved |
| 9 | SWC_CENTER_XY | R/W | 1 | Force source active window to locate at center of image.<br>Please ignore the start point of SWC_X and SWC_Y<br>(Sync. register)<br>0: Disable<br>1: Enable |
| 8 | VP_RANGE | R/W | 0 | Data range of input video<br>0: 16 ← Y ← 235<br>16 ← Cb/Cr ← 240<br>1: 1 ← Y/Cb/Cr ← 254 |
| 7 | US_SRC_SEL | R/W | 0 | Select source video data from other IP by local video I/F (Sync. register)<br>0: Source from the public video port<br>1: Source from the local video I/F |
| [6:4] | - | - | - | Reserved |
| 3 | IMG_SRC_SEL | R/W | 0 | Global source image selection<br>0: Source image is from the external image/video.<br>1: Source image is from the memory. |
| [2:0] | - | - | - | Reserved |

### 22.5.2.12 FMRATE (Offset: 0x0040)

**Table 22-13. FMRATE Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | STEP_CAP_FIRE | R/W | 0 | Fire to capture frame by the method which specified in bit 24 ~bit 27 (FIELD_XXX)<br>This bit will be set after user trigger and keep it until the end of specified frame capture (Auto clear) |
| [30:28] | - | - | - | Reserved |
| 27 | FIELD_ATRB | R/W | 0 | Video port to recognize field attribute or not<br>0: Ignore the field attribute<br>1: Enable the video port to recognize the field attribute |
| 26 | - | - | - | Reserved |
| 25 | FIELD_PAIR_EN | R/W | 0 | Get one field/frame or field-pair per fire of one stepping-frame capture<br>0: One field/frame capture<br>1: Get one field pair per fire of one stepping-frame capture |
| 24 | FIELD_SEQ | R/W | 0 | Path capture sequence by odd or even first.<br>Do not care for the progressive video<br>0: Odd first<br>1: Even first |
| [23;21] | - | - | - | Reserved |
| [20:16] | FM_RATE | R/W | 0 | The frame interval for path one to pass one frame (Sync. register)<br>0_0000: Do not skip any frame<br>0_0001 ~ 1_1110: Number of frame interval between 1 ~ 30<br>1_1111: Block all frame |
| [15:0] | - | - | - | Reserved |

### 22.5.2.13 SRCIF (Offset: 0x0044)

**Table 22-14. SRCIF Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | - | - | - | Reserved |
| [23:20] | - | - | - | Reserved |
| 19 | VP_PROG | R/W | 0 | Define CCIR 656 as progressive or interlace mode<br>0: Interlace mode<br>1: Progressive mode |
| [18:17] | - | - | - | Reserved |
| 16 | VP_F_DEF | R/W | 0 | Frame start of the video port is determined by the F or V bit at EAV/SAV<br>0: Frame start will be determined by the change of the EAV/SAV F bit for the interlace video or by the change of the EAV/SAV V bit for the progressive video. |
| [15:0] | | | | Reserved |

### 22.5.2.14 SRCSIZE0 (Offset: 0x0050)

**Table 22-15. SRCSIZE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | VP_SRC_HEIGHT | R/W | 0 | Video port image source height (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:0] | VP_SRC_WIDTH | R/W | 0 | Video port image source width (Sync. register) |

### 22.5.2.15 SRCSIZE1 (Offset: 0x0064)

**Table 22-16. SRCSIZE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | SWC_Y | R/W | 0 | Y-coordinate of top-left corner for the source active window<br>In multiples of 2 (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:0] | SWC_X | R/W | 0 | X-coordinate of top-left corner for the source active window<br>In multiples of 2 (Sync. register) |

### 22.5.2.16 SRCSIZE2 (Offset: 0x0068)

**Table 22-17. SRCSIZE2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | SWC_HEIGHT | R/W | 0 | Image source active window height<br>In multiples of 2 (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:0] | SWC_WIDTH | R/W | 0 | Image source active window width<br>In multiples of 4 (Sync. register) |

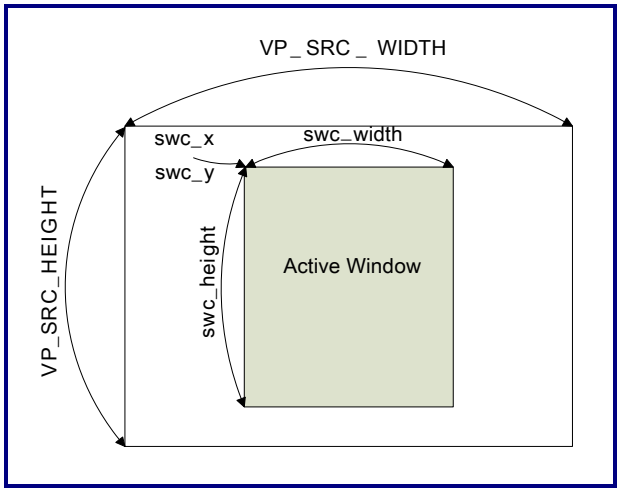



**Figure 22-7. Source Active Window**

### 22.5.2.17 SDSIZE (Offset: 0x006c)

**Table 22-18. SDSIZE Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| [28:16] | SDO_WIDTH | R | 0x00 | Size down the output image width |
| [15:13] | - | - | - | Reserved |
| [12:0] | SDO_HEIGHT | R | 0x00 | Size down the output image height |

### 22.5.2.18 IEGAIN (Offset: 0x0078)

**Table 22-19. IEGAIN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | IE_CR_GAIN- | R/W | 0x80- | Gain of chroma Cr |
| [23:16] | IE_CB_GAIN- | R/W | 0x80- | Gain of chroma Cb |
| [15:8] | - | - | - | Reserved |
| [7:0] | IE_Y_GAIN | R/W | 0x80 | Gain of luma (Sync. register) |

### 22.5.2.19 IECTRL (Offset: 0x007C)

**Table 22-20. IECTRL Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:2] | - | - | - | Reserved |
| 1 | IE_MONO_EN- | R/W | 0- | 0: Disable mono color effect<br>1: enable mono color effect |
| 0 | CS_INTACT_EN | R/W | 1- | 0: Disable the affection of contrast to saturation<br>1: Enable the affection of contrast to saturation |

### 22.5.2.20 YCPT0 (Offset: 0x00A0)

**Table 22-21. YCPT0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | YC_PT3 | R/W | 0x18 | Start point of luma mapping curve segment 3 (Sync. register) |
| [23:16] | YC_PT2 | R/W | 0x10 | Start point of luma mapping curve segment 2 (Sync. register) |
| [15:8] | YC_PT1 | R/W | 0x08 | Start point of luma mapping curve segment 1 (Sync. register) |
| [7:0] | - | - | - | - |

### 22.5.2.21 YCPT1 (Offset: 0x00A4)

**Table 22-22. YCPT1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | YC_PT7 | R/W | 0x60 | Start point of luma mapping curve segment 7 (Sync. register) |
| [23:16] | YC_PT6 | R/W | 0x40 | Start point of luma mapping curve segment 6 (Sync. register) |
| [15:8] | YC_PT5 | R/W | 0x30 | Start point of luma mapping curve segment 5 (Sync. register) |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [7:0] | YC_PT4 | R/W | 0x20 | Start point of luma mapping curve segment 4 (Sync. register) |

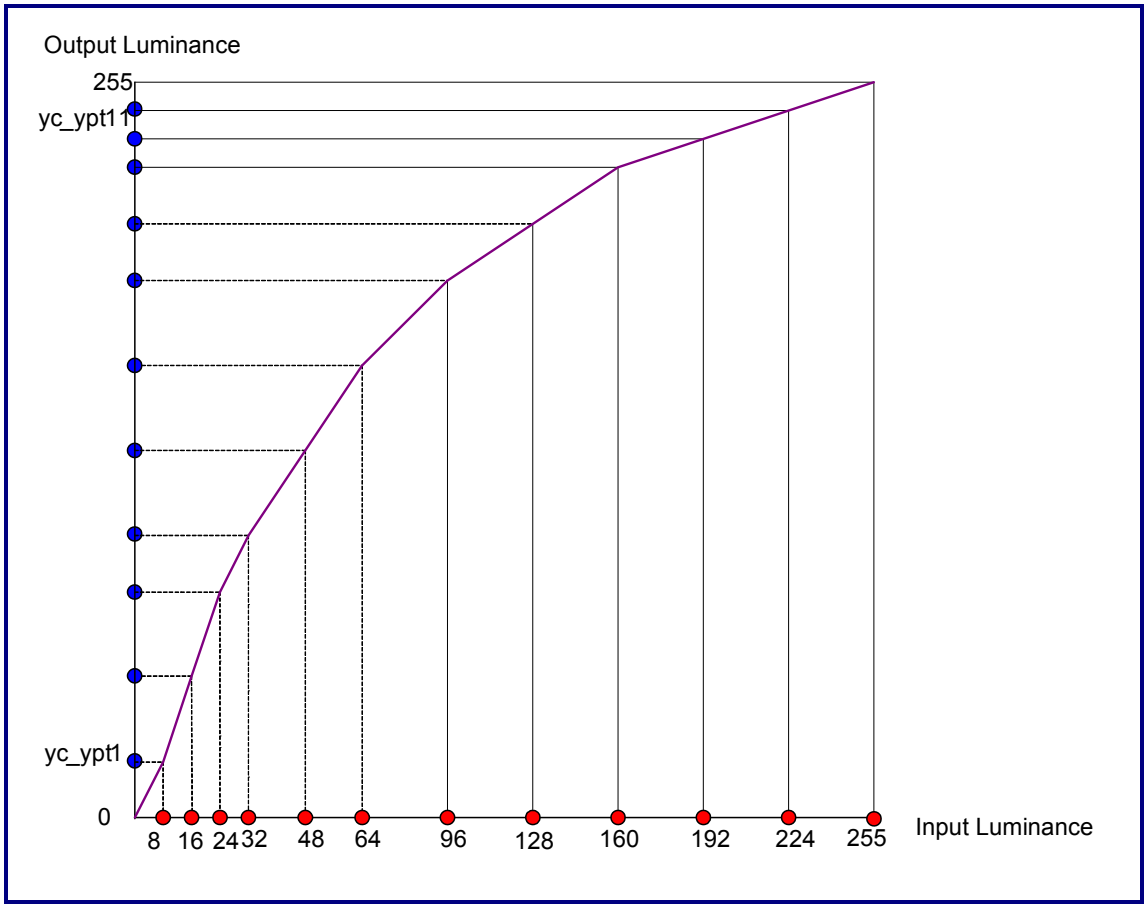



**Figure 22-8. Luma Curve**

## 22.5.2.22 YCPT2 (Offset: 0x00A8)

**Table 22-23. YCPT2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:24] | YC_PT11 | R/W | 0xE0 | Start point of segment 11 of the luma mapping curve (Sync. register) |
| [23:16] | YC_PT10 | R/W | 0xC0 | Start point of segment 10 of the luma mapping curve (Sync. register) |
| [15:8] | YC_PT9 | R/W | 0xA0 | Start point of segment 9 of the luma mapping curve (Sync. register) |
| [7:0] | YC_PT8 | R/W | 0x80 | Start point of segment 8 of the luma mapping curve (Sync. register) |

### 22.5.2.23 YCSLP0 (Offset: 0x00AC)

**Table 22-24. YCSLP0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | YC_SLOPE1 | R/W | 0x40 | Slope of luma mapping curve segment 1 (Sync. register)<br>Format: 3.6 |
| [15:9] | - | - | - | Reserved |
| [8:0] | YC_SLOPE0 | R/W | 0x40 | Slope of luma mapping curve segment 0 (Sync. register)<br>Format: 3.6 |

### 22.5.2.24 YCSLP1 (Offset: 0x00B0)

**Table 22-25. YCSLP1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | YC_SLOPE3 | R/W | 0x40 | Slope of luma mapping curve segment 3 (Sync. register)<br>Format: 3.6 |
| [15:9] | - | - | - | Reserved |
| [8:0] | YC_SLOPE2 | R/W | 0x40 | Slope of luma mapping curve segment 2 (Sync. register)<br>Format: 3.6 |

### 22.5.2.25 YCSLP2 (Offset: 0x00B4)

**Table 22-26. YCSLP2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | YC_SLOPE5 | R/W | 0x40 | Slope of luma mapping curve segment 5 (Sync. register)<br>Format: 3.6 |
| [15:9] | - | - | - | Reserved |
| [8:0] | YC_SLOPE4 | R/W | 0x40 | Slope of luma mapping curve segment 4 (Sync. register)<br>Format: 3.6 |

### 22.5.2.26 YCSLP3 (Offset: 0x00B8)

**Table 22-27. YCSLP3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | YC_SLOPE7 | R/W | 0x40 | Slope of luma mapping curve segment 7 (Sync. register)<br>Format: 3.6 |
| [15:9] | - | - | - | Reserved |
| [8:0] | YC_SLOPE6 | R/W | 0x40 | Slope of luma mapping curve segment 6 (Sync. register)<br>Format: 3.6 |

### 22.5.2.27 YCSLP4 (Offset: 0x00BC)

**Table 22-28. YCSLP1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | YC_SLOPE9 | R/W | 0x40 | Slope of luma mapping curve segment 9 (Sync. register)<br>Format: 3.6 |
| [15:9] | - | - | - | Reserved |
| [8:0] | YC_SLOPE8 | R/W | 0x40 | Slope of luma mapping curve segment 8 (Sync. register)<br>Format: 3.6 |

### 22.5.2.28 YCSLP5 (Offset: 0x00C0)

**Table 22-29. YCSLP5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| [24:16] | YC_SLOPE11 | R/W | 0x40 | Slope of segment 11 of the luma mapping curve (Sync. register)<br>Format: 3.6 |
| [15:9] | - | - | - | Reserved |
| [8:0] | YC_SLOPE10 | R/W | 0x40 | Slope of segment 10 of the luma mapping curve (Sync. register)<br>Format: 3.6 |

### 22.5.2.29 YLEVEL (Offset: 0x00C4)

**Table 22-30. YLEVEL Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:8] | - | - | - | Reserved |
| 7 | YSHIFT_SIGN | R/W | 0 | Sign bit of YSHIFT (Sync. register)<br>0: YSHIFT is a positive number.<br>1: YSHIFT is a negative number. |
| [6:5] | - | - | - | Reserved |
| [4:0] | YSHIFT | R/W | 0 | Brightness shift level (Sync. register) |

### 22.5.2.30 MXCOE0 (Offset: 0x00C8)

**Table 22-31. MXCOE0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | MX01_SIGN | R/W | 0 | Sign bit of MX01 |
| [30:27] | - | - | - | Reserved |
| [26:16] | MX01 | R/W | 0 | Color Matrix coefficient 01<br>Format: 1.10 |
| 15 | MX00_SIGN | R/W | 0 | Sign bit of MX00 |
| [14:11] | - | - | - | Reserved |
| [10:0] | MX00 | R/W | 0 | Color Matrix coefficient 00<br>Format: 1.10 |

### 22.5.2.31 MXCOE1 (Offset: 0x00CC)

**Table 22-32. MXCOE1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | MX11_SIGN | R/W | 0 | Sign bit of MX11 |
| [30:27] | - | - | - | Reserved |
| [26:16] | MX11 | R/W | 0 | Color Matrix coefficient 11<br>Format: 1.10 |
| 15 | MX10_SIGN | R/W | 0 | Sign bit of MX10 |
| [14:11] | - | - | - | Reserved |
| [10:0] | MX10 | R/W | 0 | Color Matrix coefficient 10<br>Format: 1.10 |

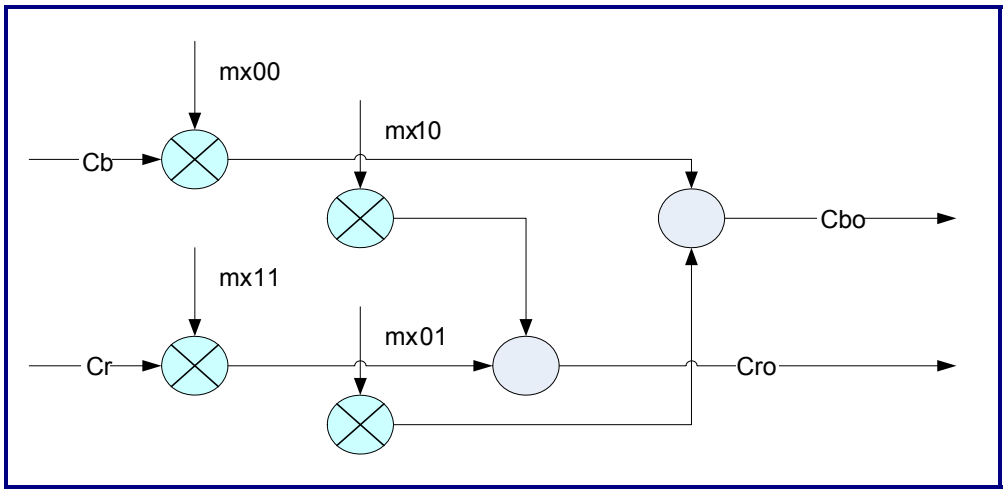



**Figure 22-9. Color Matrix**

## 22.5.2.32 SPTH (Offset: 0x00D0)

**Table 22-33. SPTH Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | SP_K2 | R/W | 0x02 | Sharpness gain factor 2<br>Format: 2.2 (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | SP_K1 | R/W | 0x04 | Sharpness gain factor 1<br>Format: 2.2 (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | SP_TH2 | R/W | 0 | Sharpness threshold 2<br>Format: 7 (Sync. register) |
| 7 | - | - | - | Reserved |
| [6:0] | SP_TH1 | R/W | 0 | Sharpness threshold 1<br>Format: 7 (Sync. register) |

### 22.5.2.33 NLRSDIF (Offset: 0x00DC)

**Table 22-34. NLRSDIF Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:16] | - | - | - | Reserved |
| [15:14] | - | - | - | Reserved |
| [13:8] | RS_DIF1_PIX | R/W | 0x00 | Different number of pixels in large window range 1 for non-linear resizing.<br>In the multiples of 2 |
| [7:6] | - | - | - | Reserved |
| [5:0] | RS_DIF0_PIX | R/W | 0x00 | Different number of pixel in large window range 0 for non-linear resizing.<br>In the multiples of 2 |

### 22.5.2.34 DDMA0 (Offset: 0x0200)

**Table 22-35. DDMA0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:4] | | | | Reserved |
| [3:2] | DMA_RLEN | R/W | 0 | DMA read burst length<br>00: 8 data package for one burst<br>01: 16 data package for one burst<br>10 ~ 11: Reserved |
| [1:0] | DMA_WLEN | R/W | 0 | DMA write burst length<br>00: 8 data package for one burst<br>01: 16 data package for one burst<br>10 ~ 11: Reserved |

### 22.5.2.35 UDMA0 (Offset: 0x0204)

**Table 22-36. UDMA0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | - | - | - | Reserved |
| [30:28] | DMA_FIFO_WM | - | 4 | Low watermark of available DMA FIFO space |
| 27 | - | - | - | Reserved |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [26:24] | HRESP_DLY | R/W | 0 | AHB retry/split delay clock = 8 + pv_hresp_dly |
| [23:17] | - | - | - | Reserved |
| 16 | DMA_MST_EN | R/W | 0 | Multi-start address enable for DMA write<br>0: Enable frame numbers 0 start address<br>1: Enable frame numbers 0, 1 start address |
| [15:14] | - | - | - | Reserved |
| [13:12] | DMA_ORDER | R/W | 0 | YCbCr 4:2:2/4:2:0 data organization in three memory buffers<br>00: Sequential order<br>01: 4 x 4 macro block order<br>10: 8 x 8 macro block order<br>11: 16 x 16 macro block order |
| [11:10] | - | - | - | Reserved |
| [9:8] | PDMA_SPLIT | R/W | 0 | YCbCr data arrangement in memory<br>00: Reserved<br>01: Write three color components into three different memory buffers<br>10: Write Y into one memory buffer and Cb/Cr into another memory buffer |
| [7:6] | DMA_CHROMA_WLEN | R/W | 0 | Cb/Cr burst write length<br>00: 1 data package for one burst<br>01: 2 data package for one burst<br>10: 4 data package for one burst<br>11: 8 data package for one burst |
| [5:4] | DMA_LUMA_WLEN | R/W | 0 | Y burst write length<br>00: 4 data package for one burst<br>01: 8 data package for one burst<br>10: 12 data package for one burst<br>11: 16 data package for one burst |
| [3:2] | - | - | - | Reserved |
| [1:0] | DMA_WLEN | R/W | 0 | DMA write burst length<br>00: 8 data package for one burst<br>01: 16 data package for one burst<br>10 ~ 11: Reserved |

### 22.5.2.36 MEMSRC0 (Offset: 0x0208)

**Table 22-37. MEMSRC0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| 28 | MEM_SRC_FIELD | R/W | 0 | Odd or even field for current memory source image<br>0: Odd field<br>1: Even field |
| 27 | - | - | - | Reserved |
| [26:24] | MEM_SRC_HBLANK | R/W | 3 | Pixel number of horizontal line blank for memory source image |
| [23:5] | - | - | - | Reserved |
| 4 | MEM_SRC_RB_SWAP | R/W | 0 | Memory source image R/B or Cb/Cr swap<br>Only for RGB565 or YCbCr 4:2:2 format<br>0: Sequence of R/Cr is ahead of B/Cb.<br>1: Sequence of B/Cb is ahead of R/Cr. |
| 3 | - | - | - | Reserved |
| [2:0] | MEM_SRC_FORMAT | R/W | 0 | Memory source image color format<br>000: RGB888<br>001: RGB565<br>010 ~ 011: Reserved<br>100: YCbCr 4:4:4<br>101: YCbCr 4:2:2<br>110 ~ 111: Reserved |

### 22.5.2.37 MEMSRC1 (Offset: 0x020C)

**Table 22-38. MEMSRC1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_SRC_ST | R/W | 0 | Memory source image start address<br>DWORD alignment |

### 22.5.2.38 MEMSRCPIT (Offset: 0x0210)

**Table 22-39. MEMSRCPIT Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:13] | - | - | - | Reserved |
| [12:0] | MEM_SRC_PIT | R/W | 0 | Pitch of source image for album mode. |

### 22.5.2.39 MEMSRC2 (Offset: 0x00214)

**Table 22-40. MEMSRC2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | MEM_SRC_HEIGHT | R/W | 0 | Memory source image height |
| [15:11] | - | - | - | Reserved |
| [10:0] | MEM_SRC_WIDTH | R/W | 0 | Memory source image width |

### 22.5.2.40 DMA1 (Offset: 0x021C)

**Table 22-41. DMA1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:11] | - | - | - | Reserved |
| [10:0] | SUB_WIN_PIT | R/W | 0 | Pitch of destination window for DMA |

### 22.5.2.41 MDEST0 (Offset: 0x0250)

**Table 22-42. MDEST0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST_YST0 | R/W | 0 | Destination frame buffer 0 Y start address<br>DWORD alignment |

### 22.5.2.42 MDEST1 (Offset: 0x0254)

**Table 22-43. MDEST1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST_YST1 | R/W | 0 | Destination frame buffer 1 Y start address<br>DWORD alignment |

## 22.5.3 MDEST2 (Offset: 0x0260)

**Table 22-44. MDEST2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST_CBST0 | R/W | 0 | Destination frame buffer 0 Cb start address<br>DWORD alignment |

### 22.5.3.1 MDEST3 (Offset: 0x0264)

**Table 22-45. MDEST3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST_CBST1 | R/W | 0 | Destination frame buffer 1 Cb start address<br>DWORD alignment |

### 22.5.3.2 MDEST4 (Offset: 0x0270)

**Table 22-46. MDEST4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [20:0] | MEM_DEST_CRST0 | R/W | 0 | Destination frame buffer 0 Cr start address<br>DWORD alignment |

### 22.5.3.3 MDEST5 (Offset; 0x0274)

**Table 22-47. MDEST5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:30] | - | - | - | Reserved |
| [29:0] | MEM_DEST_CRST1 | R/W | 0 | Destination frame buffer 1 Cr start address<br>DWORD alignment |

### 22.5.3.4 OSDFONT (Offset: 0x0300)

**Table 22-48. OSDFONT Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | FONT_RAM_READ | R/W | 0 | OSD font memory read/write setting, default write |
| [30:29] | - | - | - | Reserved |
| [28:16] | FONT_RAM_ADDR | R/W | 0 | OSD font address |
| [15:12] | - | - | - | Reserved |
| [11:0] | FONT_RAM_DATA | R/W | 0 | OSD font data |

### 22.5.3.5 OSDDISP (Offset: 0x0304)

**Table 22-49. OSDDISP Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | DISP_RAM_READ | R/W | 0 | Display ram read/write (Default = Write) |
| [30:22] | - | - | - | Reserved |
| [21:16] | DISP_RAM_ADDR | R/W | 0 | Display RAM address |
| [15:8] | - | - | - | Reserved |
| [7:0] | DISP_RAM_DATA | R/W | 0 | MSB[7], set 1 to select RAM |

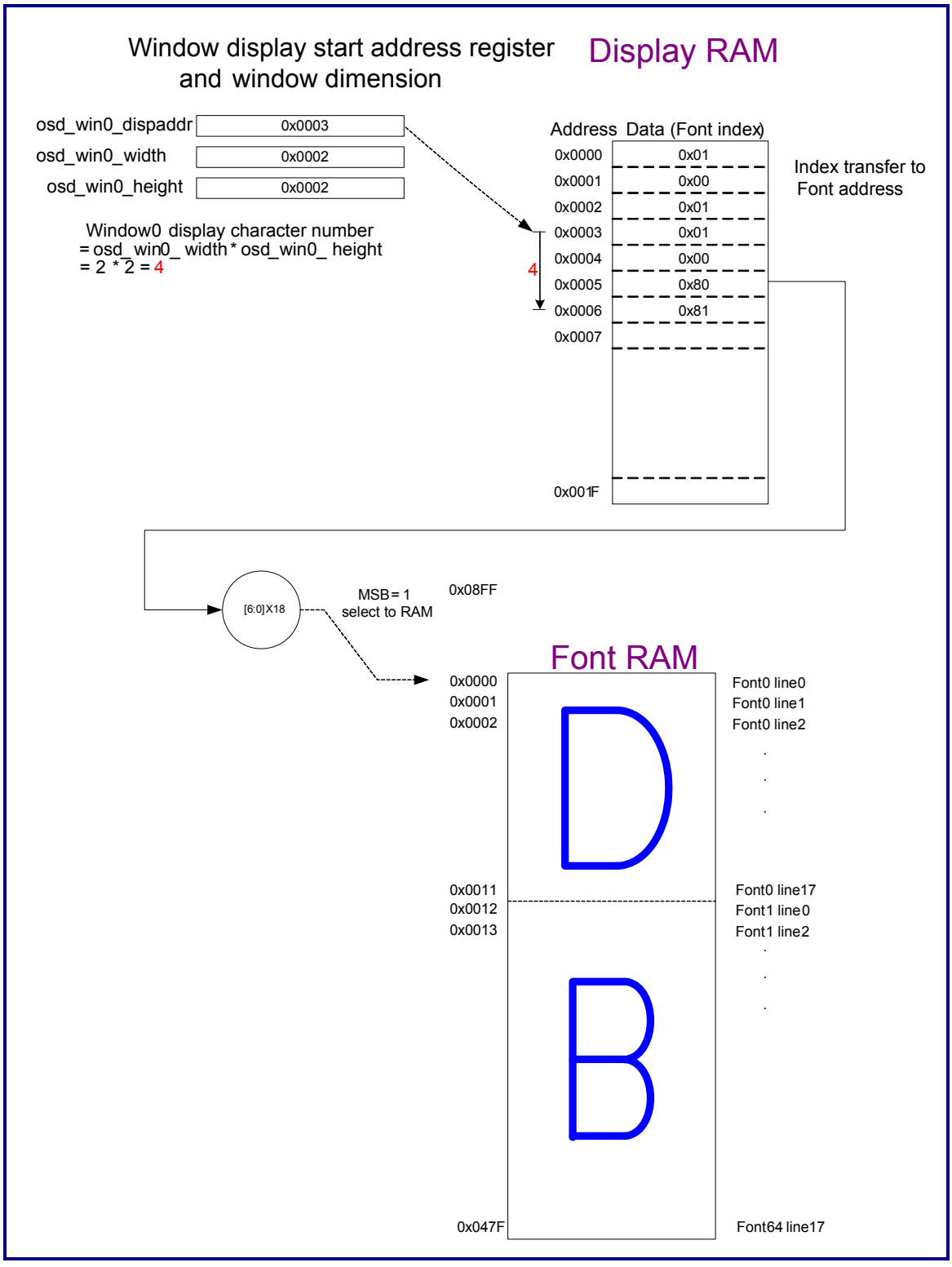



**Figure 22-10. OSD Font RAM and Display RAM**

### 22.5.3.6 OSDREAD (Offset: 0x0308)

**Table 22-50. OSDREAD Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 31 | DISP_RAM_BUSY | R | 0 | Display RAM busy |
| 30 | FONT_RAM_BUSY | R | 0 | Font memory busy |
| [29:20] | - | - | - | Reserve |
| [19:12] | DISP_RAM_DATAOUT | R | 0 | Display RAM read data out |
| [11:0] | FONT_RAM_DATAOUT | R | 0 | Font memory read data out |

### 22.5.3.7 OSDEN (Offset: 0x030C)

**Table 22-51. OSDEN Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:17] | - | - | - | Reserved |
| 16 | OSD_EDGE_SMTH_EN | R/W | 0 | Edge smooth enable function<br>0: Disable<br>1: Enable |
| [15:4] | - | - | - | Reserved |
| 3 | OSD_WIN3_EN | R/W | 0 | Window 3 enable (Sync. register) |
| 2 | OSD_WIN2_EN | R/W | 0 | Window 2 enable (Sync. register) |
| 1 | OSD_WIN1_EN | R/W | 0 | Window 1 enable (Sync. register) |
| 0 | OSD_WIN0_EN | R/W | 0 | Window 0 enable (Sync. register) |

### 22.5.3.8 OSDPAT0 (Offset: 0x0310)

**Table 22-52. OSDPAT0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | OSD_PAL0_Y | R/W | 0 | Palette color 0 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_PAL0_CB | R/W | 0 | Palette color 0 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | OSD_PAL0_CR | R/W | 0 | Palette color 0 Cr |

### 22.5.3.9 OSDPAT1 (Offset: 0x0314)

**Table 22-53. OSDPAT1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | OSD_PAL1_Y | R/W | 0 | Palette color 1 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_PAL1_CB | R/W | 0 | Palette color 1 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | OSD_PAL1_CR | R/W | 0 | Palette color 1 Cr |

### 22.5.3.10 OSDPALT2 (Offset: 0x0318)

**Table 22-54. OSDPAT2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | OSD_PAL2_Y | R/W | 0 | Palette color 2 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_PAL2_CB | R/W | 0 | Palette color 2 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | OSD_PAL2_CR | R/W | 0 | Palette color 2 Cr |

### 22.5.3.11 OSDPAT3 (Offset: 0x031C)

**Table 22-55. OSDPAT3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | OSD_PAL3_Y | R/W | 0 | Palette color 3 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_PAL3_CB | R/W | 0 | Palette color 3 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | OSD_PAL3_CR | R/W | 0 | Palette color 3 Cr |

### 22.5.3.12 OSDPAT4 (Offset: 0x0320)

**Table 22-56. OSDPAT4 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | OSD_PAL4_Y | R/W | 0 | Palette color 4 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_PAL4_CB | R/W | 0 | Palette color 4 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | OSD_PAL4_CR | R/W | 0 | Palette color 4 Cr |

### 22.5.3.13 OSDPAT5 (Offset: 0x0324)

**Table 22-57. OSDPAT5 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | OSD_PAL5_Y | R/W | 0 | Palette color 5 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_PAL5_CB | R/W | 0 | Palette color 5 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | OSD_PAL5_CR | R/W | 0 | Palette color 5 Cr |

### 22.5.3.14 OSDPAT6 (Offset: 0x0328)

**Table 22-58. OSDPAT6 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:22] | - | - | - | Reserved |
| [21:16] | OSD_PAL6_Y | R/W | 0 | Palette color 6 Y |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_PAL6_CB | R/W | 0 | Palette color 6 Cb |
| [7:5] | - | - | - | Reserved |
| [4:0] | OSD_PAL6_CR | R/W | 0 | Palette color 6 Cr |

### 22.5.3.15 OSDCOR0 (Offset: 0x0330)

**Table 22-59. OSDCOR0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | OSD_WIN0_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | OSD_WIN0_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | | | | Reserved |
| [2:0] | OSD_WIN0_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

### 22.5.3.16 OSDWSZ0 (Offset: 0x0334)

**Table 22-60. OSDWSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | OSD_WIN0_DISPADDR | R/W | 0 | Window0 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | OSD_WIN0_HEIGHT | R/W | 0 | Window0 dimension height (Sync. register)<br>Window 0 Height =<br>(OSD_WIN0_FONT_H + OSD_WIN0_FSPACE_ROW) x OSD_WIN0_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | OSD_WIN0_WIDTH | R/W | 0 | Window0 dimension width (Sync. register)<br>Window 0 Width =<br>(OSD_WIN0_FONT_W + OSD_WIN0_FSPACE_COL) x OSD_WIN0_WIDTH |

### 22.5.3.17 OSDSSZ0 (Offset: 0x0338)

**Table 22-61. OSDSSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [26:16] | OSD_WIN0_POS_Y | R/W | 0 | Window 0 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | OSD_WIN0_POS_X | R/W | 0 | Window 0 start point X (Sync. register) |

### 22.5.3.18 OSDFSZ0 (Offset: 0x033C)

**Table 22-62. OSDFSZ0 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [27:24] | OSD_WIN0_FSPACE_ROW | R/W | 0 | Window 0 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | OSD_WIN0_FSPACE_COL | R/W | 0 | Window 0 font column space (Sync. register) |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_WIN0_FONT_H | R/W | 0x8 | Window 0 font height 7 ~ 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | OSD_WIN0_FONT_W | R/W | 0xC | Window 0 font width 7 ~ 12 (Sync. register) |

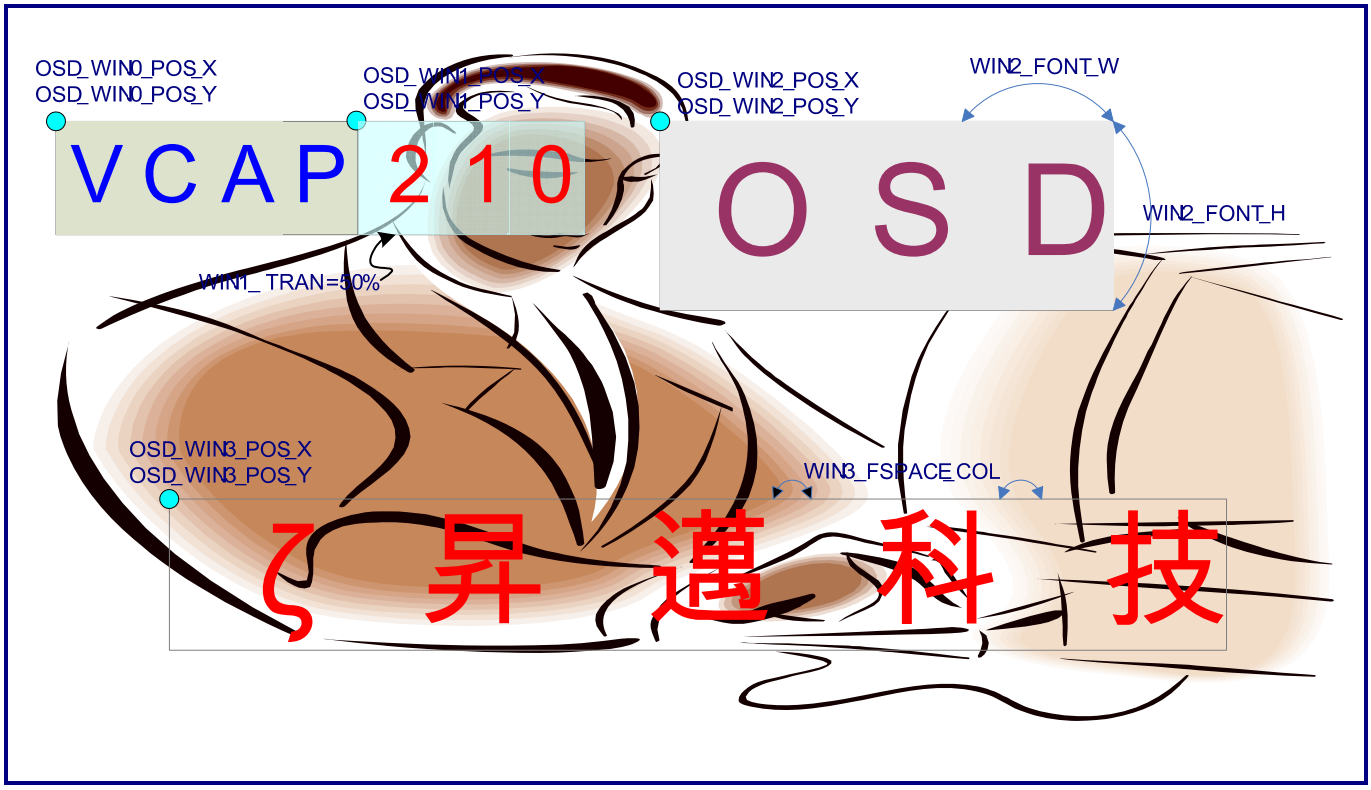



**Figure 22-11. OSD Example**

### 22.5.3.19 OSDCOR1 (Offset: 0x0340)

**Table 22-63. OSDCOR1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | OSD_WIN1_TRAN | R/W | 0 | Window font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | OSD_WIN1_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| [2:0] | OSD_WIN1_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

### 22.5.3.20 OSDWSZ1 (Offset: 0x0344)

**Table 22-64. OSDWSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | OSD_WIN1_DISPADDR | R/W | 0 | Window 1 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | OSD_WIN1_HEIGHT | R/W | 0 | Window 1 dimension height (Sync. register)<br>Window 1 height =<br>(OSD_WIN1_FONT_H + OSD_WIN1_FSPACE_ROW) x OSD_WIN1_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | OSD_WIN1_WIDTH | R/W | 0 | Window 1 dimension width (Sync. register)<br>(OSD_WIN1_FONT_W + OSD_WIN1_FSPACE_COL) x OSD_WIN1_WIDTH |

### 22.5.3.21 OSDSSZ1 (Offset: 0x0348)

**Table 22-65. OSDSSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [26:16] | OSD_WIN1_POS_Y | R/W | 0 | Window 1 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | OSD_WIN1_POS_X | R/W | 0 | Window 1 start point X (Sync. register) |

### 22.5.3.22 OSDFSZ1 (Offset: 0x034C)

**Table 22-66. OSDFSZ1 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | OSD_WIN1_FSPACE_ROW | R/W | 0 | Window 1 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | OSD_WIN1_FSPACE_COL | R/W | 0 | Window 1 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_WIN1_FONT_H | R/W | 0x8 | Window 1 font height 7 ~ 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | OSD_WIN1_FONT_W | R/W | 0xC | Window 1 font width 7 ~ 12 (Sync. register) |

### 22.5.3.23 OSDCOR2 (Offset: 0x0350)

**Table 22-67. OSDCOR2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | OSD_WIN2_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | OSD_WIN2_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| [2:0] | OSD_WIN2_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

### 22.5.3.24 OSDWSZ2 (Offset: 0x0354)

**Table 22-68. OSDWSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | OSD_WIN2_DISPADDR | R/W | 0 | Window 2 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | OSD_WIN2_HEIGHT | R/W | 0 | Window 2 dimension height (Sync. register)<br>Window 2 height =<br>(OSD_WIN2_FONT_H + OSD_WIN2_FSPACE_ROW) x OSD_WIN2_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | OSD_WIN2_WIDTH | R/W | 0 | Window 2 dimension width (Sync. register)<br>(OSD_WIN2_FONT_W + OSD_WIN2_FSPACE_COL) x OSD_WIN2_WIDTH |

### 22.5.3.25 OSDSSZ2 (Offset: 0x0358)

**Table 22-69. OSDSSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [26:16] | OSD_WIN2_POS_Y | R/W | 0 | Window 2 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | OSD_WIN2_POS_X | R/W | 0 | Window 2 start point X (Sync. register) |

### 22.5.3.26 OSDFSZ2 (Offset: 0x035C)

**Table 22-70. OSDFSZ2 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:26] | - | - | - | Reserved |
| [25:24] | OSD_WIN2_FSPACE_ROW | R/W | 0 | Window 2 font row space (Sync. register) |
| [23:18] | - | - | - | Reserved |
| [19:16] | OSD_WIN2_FSPACE_COL | R/W | 0 | Window 2 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_WIN2_FONT_H | R/W | 0 | Window 2 font height 7 ~ 12 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | OSD_WIN2_FONT_W | R/W | 0 | Window 2 font width 7 ~ 12 (Sync. register) |

### 22.5.3.27 OSDCOR3 (Offset: 0x0360)

**Table 22-71. OSDCOR3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:10] | - | - | - | Reserved |
| [9:8] | OSD_WIN3_TRAN | R/W | 0 | Window Font backward color<br>Transparency level<br>00: 0% (Normal)<br>01: 50%<br>10: 75%<br>11: 100% |
| 7 | - | - | - | Reserved |
| [6:4] | OSD_WIN3_BWCOLOR | R/W | 0 | Window font backward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |
| 3 | - | - | - | Reserved |
| [2:0] | OSD_WIN3_FWCOLOR | R/W | 0 | Window font forward color index<br>000: Palette color No.0<br>001: Palette color No.1<br>010: Palette color No.2<br>011: Palette color No.3<br>100: Palette color No.4<br>101: Palette color No.5<br>110: Palette color No.6<br>111: Inverse background color |

### 22.5.3.28 OSDWSZ3 (Offset: 0x0364)

**Table 22-72. OSDWSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:23] | - | - | - | Reserved |
| [22:16] | OSD_WIN3_DISPADDR | R/W | 0 | Window 3 character display start address (Sync. register) |
| 15 | - | - | - | Reserved |
| [14:8] | OSD_WIN3_HEIGHT | R/W | 0 | Window 3 dimension height (Sync. register)<br>Window 3 height =<br>(OSD_WIN3_FONT_H + OSD_WIN3_FSPACE_ROW) x OSD_WIN3_HEIGHT |
| 7 | - | - | - | Reserved |
| [6:0] | OSD_WIN3_WIDTH | R/W | 0 | Window 3 dimension width (Sync. register)<br>(OSD_WIN3_FONT_W + OSD_WIN3_FSPACE_COL) x OSD_WIN3_WIDTH |

### 22.5.3.29 OSDSSZ3 (Offset: 0x0368)

**Table 22-73. OSDSSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:27] | - | - | - | Reserved |
| [26:16] | OSD_WIN3_POS_Y | R/W | 0 | Window 3 start point Y (Sync. register) |
| [15:11] | - | - | - | Reserved |
| [10:0] | OSD_WIN3_POS_X | R/W | 0 | Window 3 start point X (Sync. register) |

### 22.5.3.30 OSDFSZ3 (Offset: 0x036C)

**Table 22-74. OSDFSZ3 Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:28] | - | - | - | Reserved |
| [27:24] | OSD_WIN3_FSPACE_ROW | R/W | 0 | Window 3 font row space (Sync. register) |
| [23:20] | - | - | - | Reserved |
| [19:16] | OSD_WIN3_FSPACE_COL | R/W | 0 | Window 3 font column space (Sync. register) |
| [15:13] | - | - | - | Reserved |
| [12:8] | OSD_WIN3_FONT_H | R/W | 0x8 | Window 3 font height 7 ~ 18 (Sync. register) |
| [7:4] | - | - | - | Reserved |
| [3:0] | OSD_WIN3_FONT_W | R/W | 0xC | Window 3 font width 7 ~ 12 (Sync. register) |

### 22.5.3.31 INTSTS (Offset: 0x03E0)

**Table 22-75. INTSTS Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| 28 | DMA_OVF_ERR | R/W | 0 | DMA write overflow error (Write one clear) |
| 27 | UAHB_ERR | R/W | 0 | AMBA AHB receives error response at the upstream path<br>(Write one clear) |
| 26 | NULL_ERR | R/W | 0 | Frame done status is not found until the start of the next frame<br>(Write one clear) |
| 25 | FD_QUE_STS | R/W | 0 | Queue of Field/Frame capture done status |
| 24 | FD_STS | R/W | 0 | One Field/Frame capture done status |
| [23:12] | - | - | - | Reserved |
| 11 | DAHB_ERR | R/W | 0 | AMBA AHB receives error response at the downstream path |
| [10:9] | - | - | - | Reserved |
| 8 | DAHB_IDLE_STS | R/W | 0 | AMBA AHB transmits one frame then return to the idle state at the downstream path |
| [7:6] | - | - | - | Reserved |
| 5 | FM_END_STS | R/W | 0 | Status of the frame end |
| 4 | FM_START_STS | R/W | 0 | Status of the frame start |
| [3:0] | - | - | - | Reserved |

### 22.5.3.32 INTMASK (Offset: 0x03E4)

**Table 22-76. INTMASK Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:29] | - | - | - | Reserved |
| 28 | DMA_OVF_MASK | R/W | 0 | Mask interrupt from DMA write overflow error<br>0: Retain DMA write overflow interrupt<br>1: Mask DMA write overflow interrupt |
| 27 | UAHB_ERR_MASK | R/W | 0 | Mask interrupt from AMBA AHB error response at upstream path<br>0: Retain AMBA AHB error interrupt at upstream path<br>1: Mask AMBA AHB error interrupt at upstream path |

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| 26 | NULL_ERR_MASK | R/W | 0 | Mask interrupt from null_err<br>0: Retain null_err interrupt source<br>1: Mask null_err interrupt source |
| 25 | - | - | - | Reserved |
| 24 | FD_STS_MASK | R/W | 0 | Mask interrupt from fd_sts<br>0: Retain fd_sts interrupt source<br>1: Mask fd_sts interrupt source |
| [23:12] | - | - | - | Reserved |
| 11 | DAHB_ERR_MASK | R/W | 0 | Mask interrupt from AMBA AHB error response at downstream path<br>0: Retain AMBA AHB error interrupt at downstream path<br>1: Mask AMBA AHB error interrupt at downstream path |
| [10:9] | - | - | - | Reserved |
| 8 | DAHB_IDLE_MASK | R/W | 0 | Mask interrupt from AHB IDLE at downstream path<br>0: Retain AHB IDLE interrupt at downstream path<br>1: Mask AHB IDLE interrupt at downstream path |
| [7:6] | - | - | - | Reserved |
| 5 | FM_END_MASK | R/W | 1 | Mask interrupt from frame end<br>0: Retain interrupt from frame end<br>1: Mask interrupt from frame end |
| 4 | FM_START_MASK | R/W | 1 | Mask interrupt from frame start<br>0: Retain interrupt from frame start<br>1: Mask interrupt from frame start |
| [3:0] | - | - | - | Reserved |

### 22.5.3.33 SCARSTS (Offset: 0x03EC)

**Table 22-77. SCARSTS Register**

| Bit | Symbol | Access | Default | Description |
|---|---|---|---|---|
| [31:25] | - | - | - | Reserved |
| 24 | CU_BUSY | R | 0 | Current CU busy status. Value “0” to indicate busy status |
| [23:19] | - | - | - | Reserved |
| [18:16] | VP_STA | - | - | Current Video port status to indicate active and blanking interval |
| [15:0] | - | - | - | Reserved |

## 22.6 Function Descriptions

### 22.6.1 Video Port

This block is used to receive video data by the ITU-R. 656 interface.

#### 22.6.1.1 ITU-R BT. 656

**Table 22-78. ITU-R BT. 656 EAV and SAV Sequence**

| | 8-bit Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Preamble | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Status Word | 1 | F | V | H | P3 | P2 | P1 | P0 |

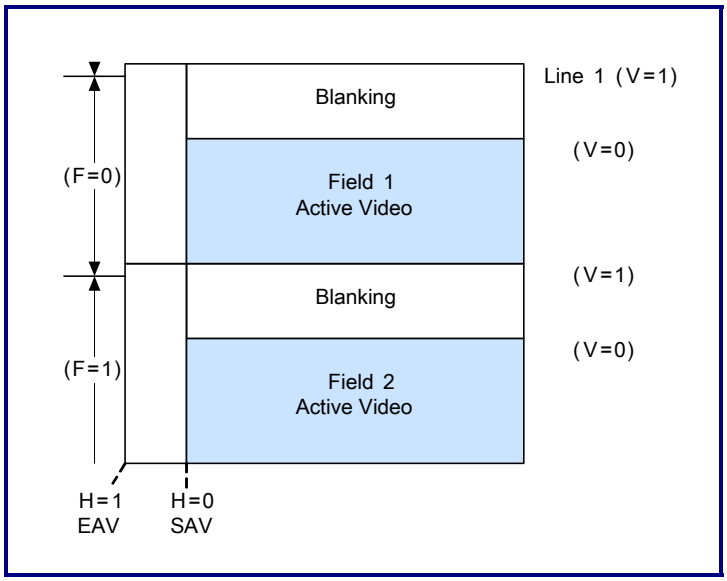



**Figure 22-12. Vertical interval of ITU-R BT. 656**

Only the SAV and EAV sequences are used to recover the video timing. Please do not make any assumption about the number of clock cycles per line or horizontal blanking interval.

### 22.6.1.2 ITU-R BT. 656 Like

| | 8-bit Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| Preamble | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Status Word | 1 | F | V | H | P3 | P2 | P1 | P0 |
| Vertical start (VS) | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Vertical end (VE) | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| Horizontal start (HS) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Horizontal end (HE) | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

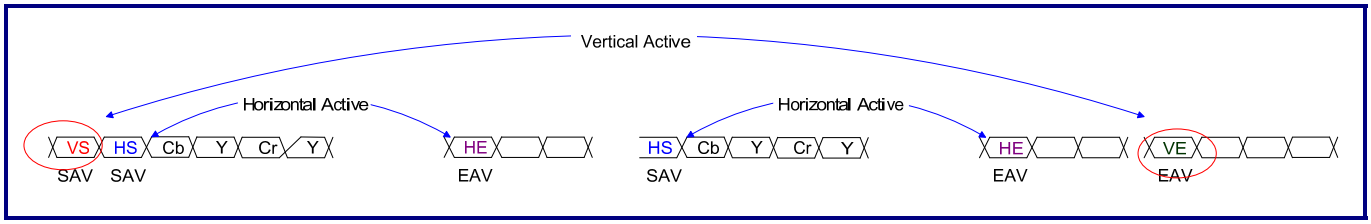



**Figure 22-13. Timing of ITU-R BT. 656 Like**

Note: There is no EAV/SAV codes for the horizontal line during the vertical blanking.

## 22.6.2 Size-down

This block is used to size-down the image with the independent integer horizontal and vertical ratio down to 1/127. The ratio of size down can be independent on two video capture paths. The line buffers are needed for the size-down algorithm to provide high quality size-down images.

## 22.6.3 OSD

This block is used to paste any character to the video capture data. Four windows with different attributes can be defined. 64 fonts with size of 12 x 18 are programmable and a maximum of 128 fonts can be shown on the display window. Adjustable font size, font space, font color, and transparency attribute are provided for variety applications.

### 22.6.4 Color Space Conversion

This block is used to convert the color space between the YCbCr and RGB domains.

### 22.6.5 Window Clip

This block is used to clip the target region from images before or after sizing down. The color border with the controllable width can be appended to the clipping window.

### 22.6.6 DMA

This block is used to send video data with any color format into any memory location by the scan-line or macro block order. The FIFO depth is configurable depending on the available bandwidth on the system.

#### 22.6.6.1 Raster Order Sequence

##### 22.6.6.1.1 RGB888

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data | B0G0R0 B1 | G1R1B2 G2 | R2B3G3 R3 |

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data (BR swap) | R0G0B0 R1 | G1B1R2 G2 | B2R3G3 B3 |

##### 22.6.6.1.2 RGB565

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data | B0G0R0 B1G1R1 | B2G2R2 B3G3R3 | B4G4R4 B5G5R5 |

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data (BR swap) | R0G0B0 R1G1B1 | R2G2B2 R3G3b3 | R4G4B4 R5G5B5 |

### 22.6.6.1.3 YCbCr 4:4:4

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data | Cb0Y0Cr0 Cb1 | Y1Cr1Cb2 Y2 | Cr2Cb3Y3 Cr3 |

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data (CbCr swap) | Cr0Y0Cb0 Cr1 | Y1Cb1Cr2 Y2 | Cb2Cr3Y3 Cb3 |

### 22.6.6.1.4 YCbCr 4:2:2

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data | Cb0Y0Cr0 Y1 | Cb2Y2Cr2 Y3 | Cb4Y4Cr4 Y5 |

| Offset | 0x0 | 0x4 | 0x8 |
|---|---|---|---|
| Data (CbCr swap) | Cr0Y0Cb0 Y1 | Cr2Y2Cb2 Y3 | Cr4Y4Cb4 Y5 |

## 22.6.6.2 Macro Block Order Sequence

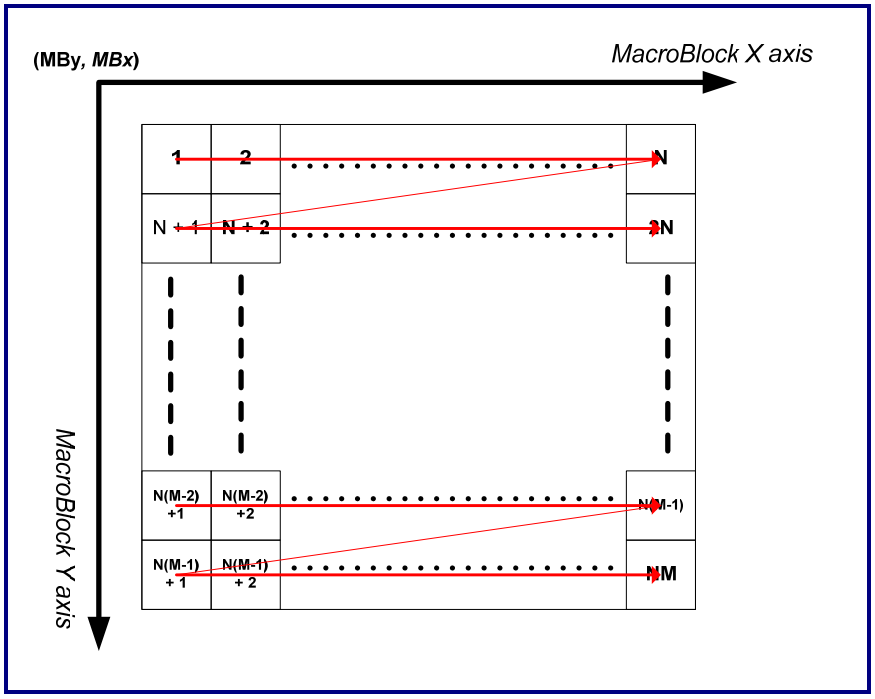



**Figure 22-14. Macro Block Order in One Image**

### 22.6.6.2.1 Macro Block 16 x 16

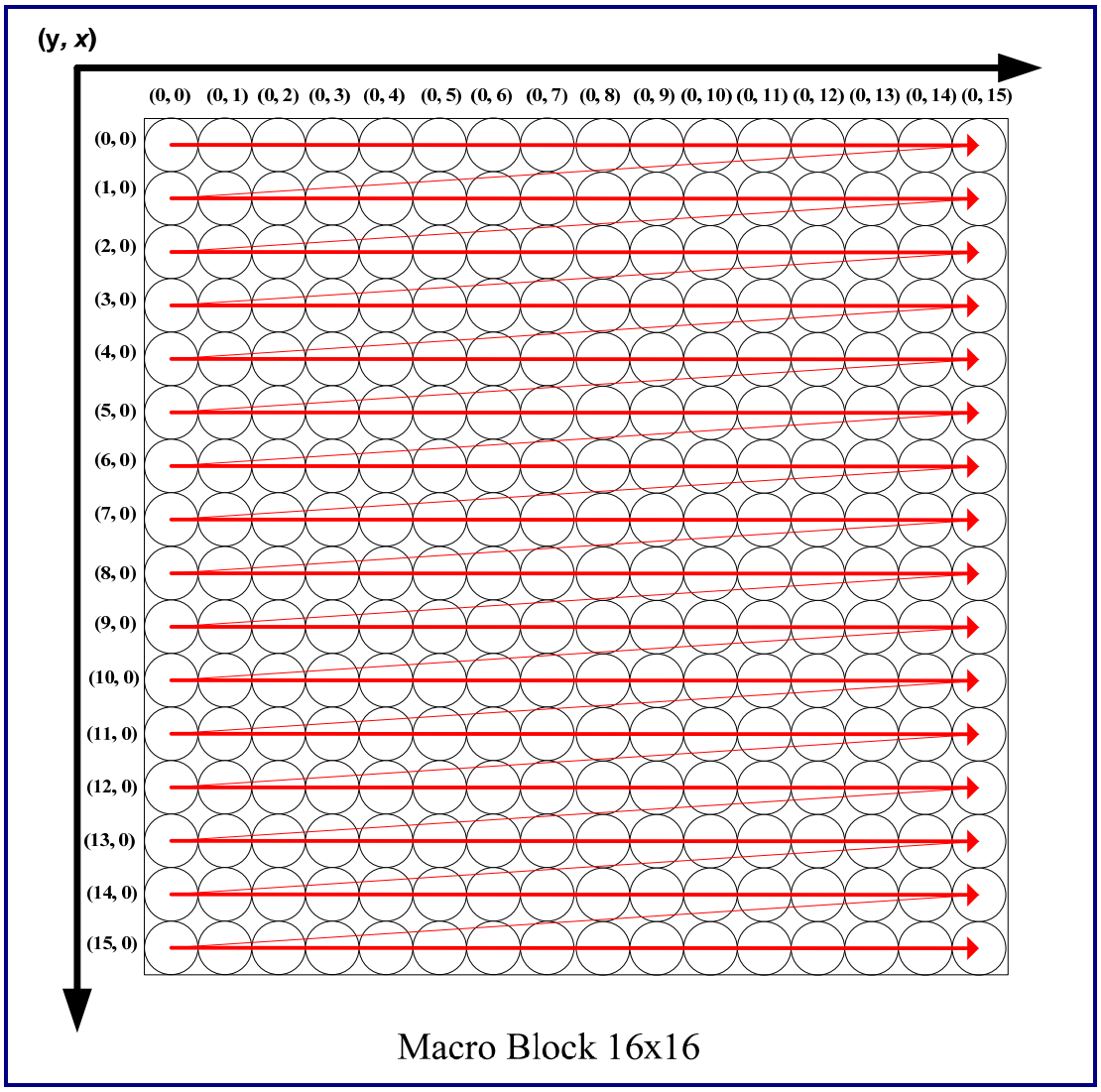



**Figure 22-15. Pixel Output Order in One 16 x 16 Macro Block**

### 22.6.6.2.2 Macro Block 8 x 8

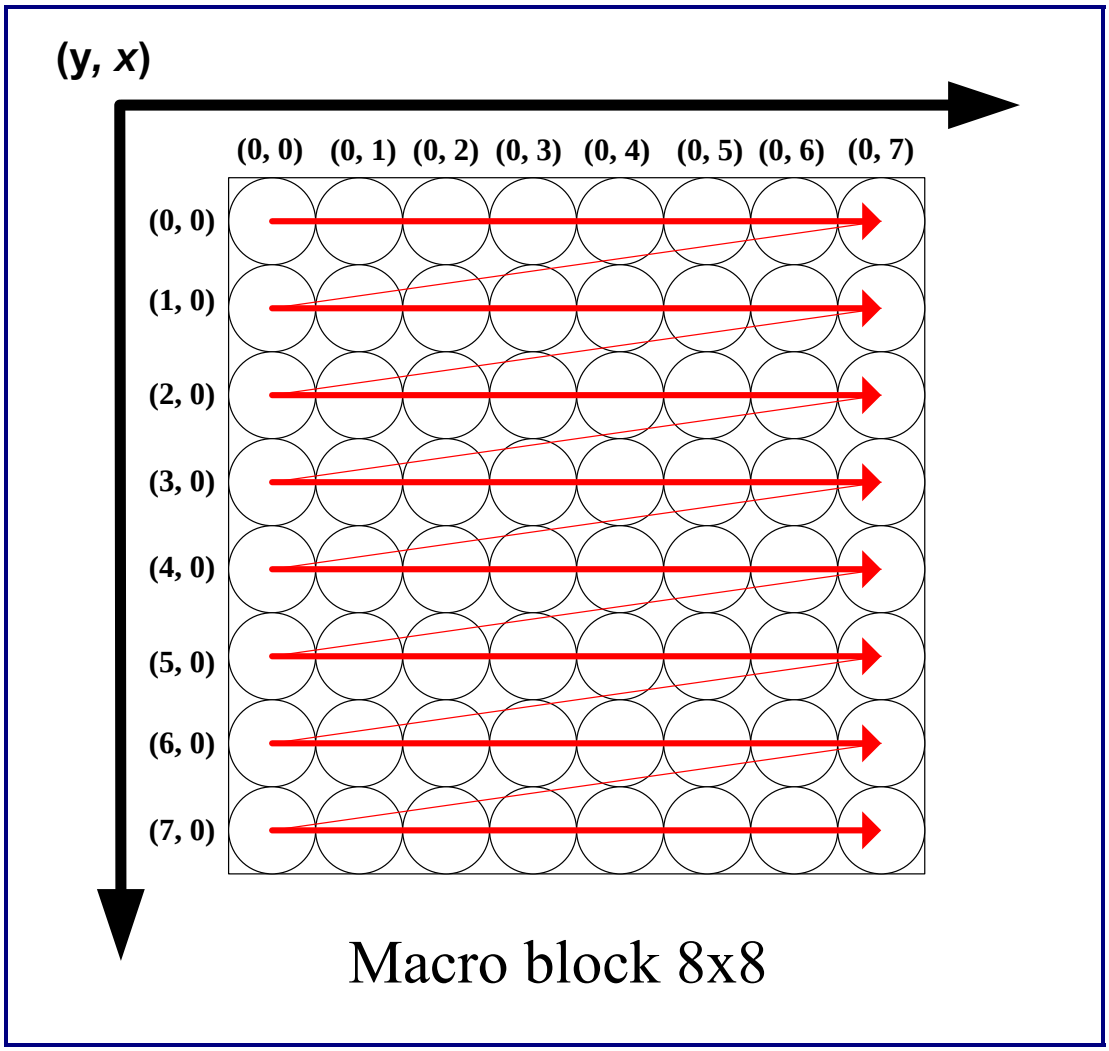



**Figure 22-16. Pixel Output Order in One 8 x 8 Macro Block**

### 22.6.6.2.3 Macro Block 4 x 4

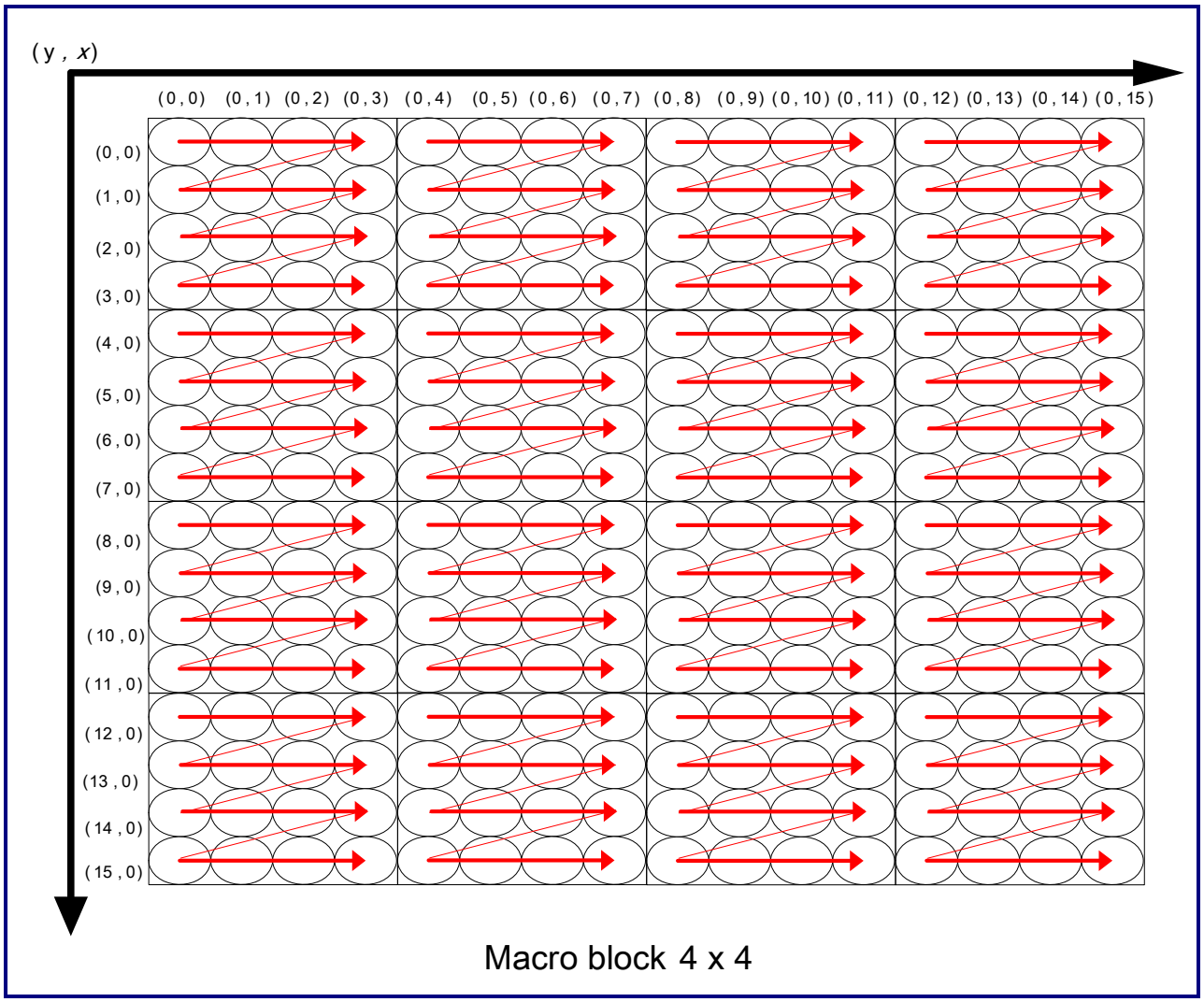



**Figure 22-17. Pixel Output Order in One 4 x 4 Macro Block**

### 22.6.6.3 Two-split Memory Block

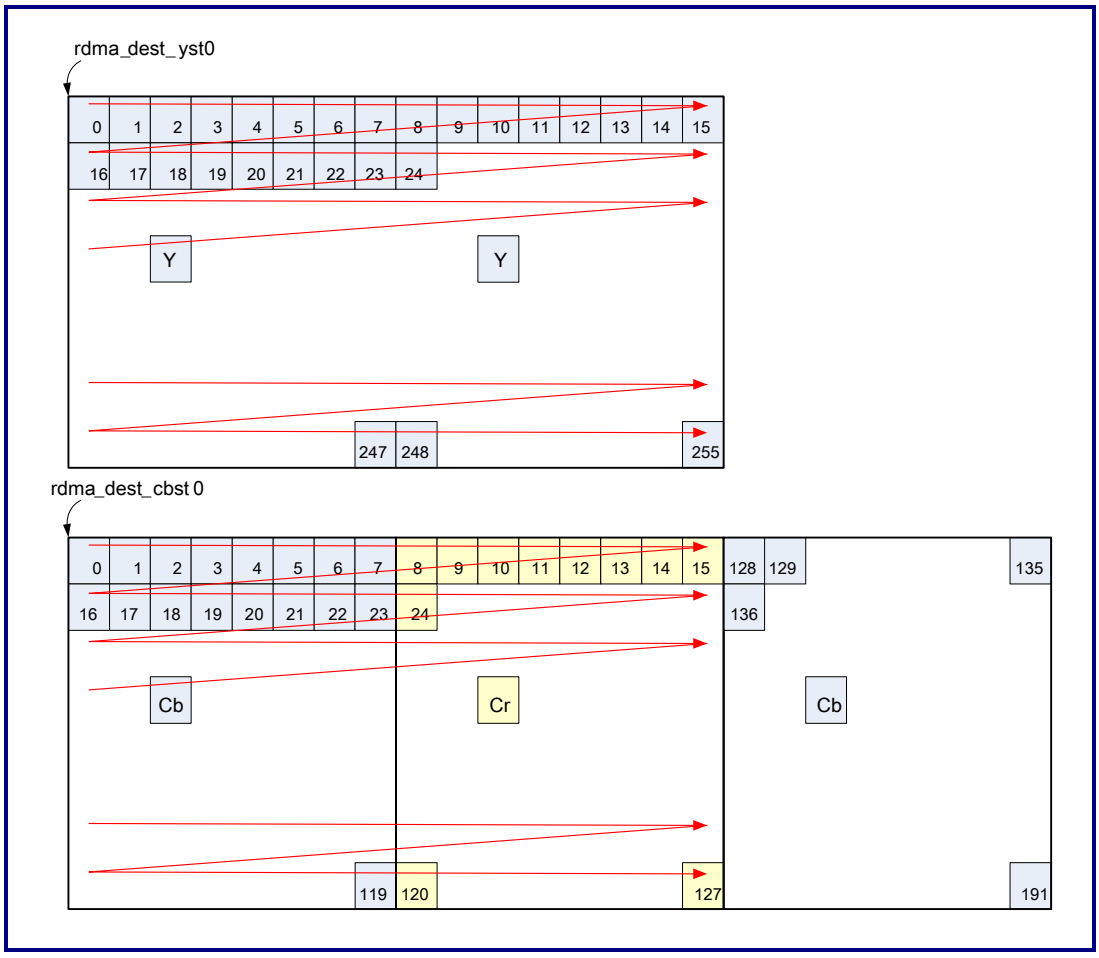



**Figure 22-18. YCbCr 4:2:0 Macro Block Order in Two-split Memory Block**

# Chapter 23

# UART Controller

This chapter contains the following sections:


- 23.1 General Description
- 23.2 Features
- 23.3 Block Diagram
- 23.4 Programming Model

## 23.1 General Description

The UART controller is a serial communication element that implements the UART operation mode and is backward compatible to 16550 to support the existing communications software.

## 23.2 Features

- High-speed NS 16C550A-compatible UART
- Programmable baud rates of up to 1152 Kbps
- Ability to add or delete the standard asynchronous communications bits (Start, stop, and parity) of the serial data
- Programmable baud rate generator to divide internal clock from 1 to ($2^{16}$-1) to generate internal 16X clock
- Fully programmable serial interface:
  - 5-bit, 6-bit, 7-bit, or 8-bit characters
  - Even, odd, and no parity detection
  - 1, 1.5, or 2 stops bit generation
- Complete status reporting capability
- Ability to generate and detect line breaks
- Fully prioritized interrupt system controls
- Separate DMA requests for transmit and receive data services
- Break, parity, overrun, framing error simulation for the UART mode
- UART provides the 16-byte transmit FIFO and 16-byte receive FIFO

## 23.3 Block Diagram

Figure 23-1 shows the block diagram of the Faraday UART controller.

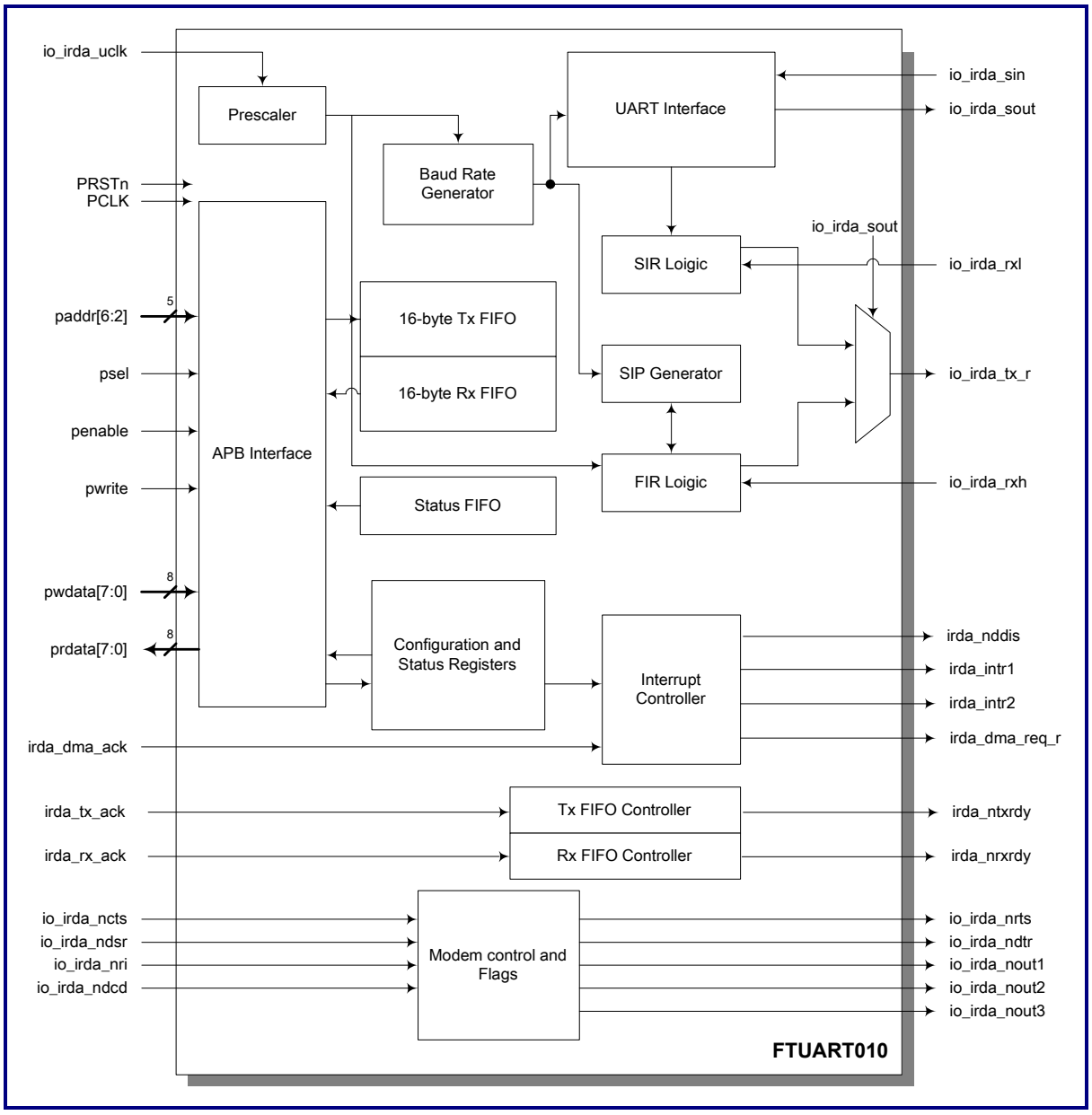



**Figure 23-1. Block Diagram of UART Controller**

## 23.4 Programming Model

### 23.4.1 Summary of IrDA Communications Controller Registers

**Table 23-1. Summary of UART Controller Registers**

| Offset | Type | Width | Name | Description | Reset Value |
|---|---|---|---|---|---|
| **UART Mode** | | | | | |
| +0x00 | R | 8 | RBR | Receiver Buffer Register | 0x00 |
| | W | 8 | THR | Transmitter Holding Register | 0x00 |
| +0x04 | R/W | 4 | IER | Interrupt Enable Register | 0x00 |
| +0x08 | R | 8 | IIR | Interrupt Identification Register | 0x01 |
| | W | | FCR | FIFO Control Register | 0x00 |
| +0x0C | R/W | 8 | LCR | Line Control Register | 0x00 |
| +0x10 | R/W | 7 | MCR | Modem Control Register | 0x00 |
| +0x14 | R | 8 | LSR | Line Status Register | 0x60 |
| | W | | TST | Testing Register | 0x00 |
| +0x18 | R | 8 | MSR | Modem Status Register | 0x00 |
| +0x1C | R/W | 8 | SPR | Scratch Pad Register | 0x00 |
| **Registers accessible when DLAB = 1** | | | | | |
| +0x00 | R/W | 8 | DLL | Baud Rate Divisor Latch Least Significant Byte | 0x01 |
| +0x04 | R/W | 8 | DLM | Baud Rate Divisor Latch Most Significant Byte | 0x00 |
| +0x08 | R/W | 5 | PSR | Prescaler Register | 0x01 |

### 23.4.2 Register Descriptions

#### 23.4.2.1 Receiver Buffer Register (RBR, Offset: 0x00 for Read)

Users can get the data by reading this read-only location. It is the data read port of the RX FIFO or 1-byte register depending on whether the FIFOs are enabled or not.

- If the FIFOs are enabled:
  - This location refers to the top of 16-byte FIFO (i.e. the next to be read).
- If the FIFOs are not enabled:
  - This location refers to a 1-byte register (The bottom word of the receive FIFO) that receives the contents of the receiver shift register once a character has been assembled.

**Table 23-2. Receiver Buffer Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| 8 | RBR | R | Receive Data Port |

### 23.4.2.2 Transmitter Holding Register (THR, Offset: 0x00 for Write)

The transmitter holding register is used to write the transmitter holding register or the transmit FIFO depending on whether FIFO is enabled or not.

- If FIFO is enabled:
  - Data written to this location is pushed onto the transmit FIFO.
- If FIFO is not enabled:
  - Data written to this location is stored in the transmitter holding register (The bottom entry of the transmit FIFO).

If the transmitted character width is less than eight bits, it must be right-justified. The left bits (i.e. MSB) are "don't care" bits. For example, with a word length of five bits, writing 0xd3 or 0xf3 will result in the transmission of a 13h character. Before writing this register, the user must ensure that FTUART010 is ready to accept data for transmission; that is, to ensure that THR Empty flag is set in the LSR (Please refer to the description of this register below).

**Table 23-3. Transmitter Holding Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| 8 | THR | W | Transmit data port |

### 23.4.2.3 Interrupt Enable Register (IER, Offset: 0x04)

This register individually enables each possible interrupt source. Writing a '1' in any of these bits enables the corresponding interrupt, while writing a '0' disables the operation. For detailed descriptions of the interrupt sources, please refer to the description of the Interrupt Identification Register (IIR) on the next page.

**Table 23-4. Interrupt Enable Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:4] | Reserved | - | - |
| 3 | MODEM Status | R/W | This bit enables the modem status interrupt when set to logic 1. |
| 2 | Receiver Line Status | R/W | This bit enables the Receiver Line Status Interrupt when set to logic 1. |
| 1 | THR Empty | R/W | This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1. |
| 0 | Receiver Data Available | R/W | This bit enables the Received Data Available Interrupt (And character reception timeout interrupts in the FIFO mode) when set to logic 1. |

### 23.4.2.4 Interrupt Identification Register (IIR, Offset: 0x08)

The main purpose of this register is to identify the interrupt of the highest priority that is currently pending. FTUART010 implements a 4-level priority encoder from the highest priority to the lowest priority, as follows:

1. Receive Line Status (Highest priority)
2. Receive Data Ready and Character Reception Timeout (Second priority)
3. Transmitter Holding Register Empty (Third priority)
4. Modem Status (Lowest priority)

Table 23-5 describes different interrupt conditions and the codes of identification, together with the reset method.

**Table 23-5. Interrupt Identification Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:6] | FIFO mode enable | R | These two bits are set when FCR[0] is set as 1. |
| 5 | Reserved | R | This bit of the IIR is always a 0. |
| 4 | Tx FIFO full | R | This bit is set as 1 when TX FIFO is full. |
| 3 | FIFO mode only | R | In the 16450 mode, this bit is 0. In the FIFO mode, this bit is set along with bit 2 when a timeout interrupt is pending. |
| [2:1] | Interrupt Identification Code | R | These bits identify the highest priority interrupt that is pending. Table 23-6 describes the different interrupt conditions and their codes of identification, together with their reset method. Please note that for an interrupt source to be considered as pending, the corresponding bit in the IER must be enabled. |

| Bit | Name | Type | Function |
|---|---|---|---|
| 0 | Interrupt Pending | R | This bit can be used in a prioritized interrupt environment to indicate whether an interrupt is pending.<br>0: An interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine.<br>1: No interrupt is pending. |

**Table 23-6. Interrupt Control Table**

| Interrupt Identification Register | | | | | Interrupt Set and Reset Functions | | |
|---|---|---|---|---|---|---|---|
| **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** | **Priority Level** | **Interrupt Type** | **Interrupt Source Description** | **Interrupt Reset Method** |
| 0 | 0 | 0 | 1 | - | None | There is no interrupt pending. | None |
| 0 | 1 | 1 | 0 | Highest | Receiver Line Status | There is an overrun error, parity error, framing error or break interrupt indication corresponding to the received data on top of the receive FIFO.<br>Please note that the FIFO error flag in LSR does not influence this interrupt, which is related only to the data on top of the Rx FIFO. This is directly related to the presence of a 1 in any of the LSR bits 1 to 4. | Read the Line Status Register (LSR) |
| 0 | 1 | 0 | 0 | Second | Received Data Ready | In the non-FIFO mode, there is received data available in the RHR register.<br>In the FIFO mode, the number of characters in the receive FIFO is equal to or greater than the trigger level programmed in FCR. The interrupt signal will stay active while the number of words in the FIFO stays higher than that value and will be cleared when the microprocessor reads the necessary words to make the number of words in the FIFO less than the trigger level.<br>Please note that this is not directly related to LSR bit 0, which always indicates that there is at least one word ready. | Read the Receiver Buffer Register (RBR) |

| Interrupt Identification Register | | | | | Interrupt Set and Reset Functions | | |
|---|---|---|---|---|---|---|---|
| **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** | **Priority Level** | **Interrupt Type** | **Interrupt Source Description** | **Interrupt Reset Method** |
| 1 | 1 | 0 | 0 | Second | Character Reception Timeout | There is at least one character in the receive FIFO and during a time corresponding to four characters at the selected baud rate, no new character has been received.<br>A FIFO timeout interrupt will occur, if the following conditions exist:<br>1. At least one character is in the FIFO.<br>2. The most recent serial character received was longer than 4 continuous character times ago (if 2 stop bits are programmed, the second one is included in this time delay).<br>3. The most recent CPU read of the FIFO was longer than four continuous characters time ago. | Read the Receiver Buffer Register (RBR)<br>Note: In the UART mode, if RX timeout is asserted, reset the FIFO will not clear it, user must read RBR |
| 0 | 0 | 1 | 0 | Third | Transmitter Holding Register Empty | In the non-FIFO mode, the 1-byte THR is empty. In FIFO mode, the complete 16-byte transmit FIFO is empty, so 1 to 16 characters can be written to THR. That is to say, THR Empty bit in LSR is one. | Write the Transmitter Holding Register (THR)<br>Alternatively, reading the Interrupt Identification Register (IIR) will also clear the interrupt if this is the interrupt type being currently indicated (this will not clear the flag in the LSR). |
| 0 | 0 | 0 | 0 | Fourth | Modem Status | A change has been detected in the Clear To Send (CTS), Data Set Ready (DSR) or Carrier Detect (CD) input lines or a trailing edge in the Ring Indicator (RI) input line. That is to say, at least one of MSR bits 0 to 3 is one. | Read the Modem Status Register (MSR) |

### 23.4.2.5 FIFO Control Register (FCR, Offset: 0x08 for Write)

This is a write-only register at the same location as IIR (IIR is a read-only register). This register is used to enable and clear FIFOs, and set the RX FIFO trigger level. If users reset FIFO, the status FIFO will not be cleared. Users must clear the status FIFO by reading the corresponding registers.

**Table 23-7. FIFO Control Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:6] | RXFIFO_TRGL | W | Used to set the trigger level of the RX FIFO interrupt. Please refer to Table 23-8. |
| [5:4] | TXFIFO_TRGL | W | Used to set the trigger level of the TX FIFO interrupt. Please refer to<br><br>Table 23-9. |
| 3 | DMA Mode | W | This bit selects the UART DMA mode. The DMA mode affects the way in which the DMA signaling outputs pins (irda_nrxrdy and irda_ntxrdy) behave. |
| 2 | TX FIFO Reset | W | Setting this bit to logic 1 clears all bytes in the TX FIFO and resets its counter logic to 0. The shift register is not cleared, so any reception active will continue.<br>This bit will automatically return to zero. |
| 1 | RX FIFO Reset | W | Setting this bit to logic 1 clears all bytes in the Rx FIFO and resets its counter logic to 0. The shift register is not cleared, so any reception active will continue. Setting this bit also clears the Status FIFO.<br>This bit will automatically return to zero. |
| 0 | FIFO Enable | W | Set this bit to logic 1 enables both the transmit and receive FIFOs (And Status FIFO). Changing this bit automatically resets both FIFOs.<br>In a FIR mode, the device driver should always set this bit as 1. |

Table 23-8. FIFO Trigger Level of the Receiver

| FCR Code | | FIFO Trigger Level of 16-byte Receiver | FIFO Trigger Level of 32-byte Receiver |
|---|---|---|---|
| Bit 7 | Bit 6 | | |
| 0 | 0 | 1 character | 1 character |
| 0 | 1 | 4 characters | 8 characters |
| 1 | 0 | 8 characters | 16 characters |
| 1 | 1 | 14 characters | 28 characters |
| FCR Code | | FIFO Trigger Level of 64-byte Receiver | FIFO Trigger Level of 128-byte Receiver |
| Bit 7 | Bit 6 | | |
| 0 | 0 | 1 character | 1 character |
| 0 | 1 | 16 characters | 32 characters |
| 1 | 0 | 32 characters | 64 characters |
| 1 | 1 | 56 characters | 120 characters |

Table 23-9. FIFO Trigger Level of the Transmitter

| FCR Code | | FIFO Trigger Level of 16-byte Transmitter | FIFO Trigger Level of 32-byte Transmitter |
|---|---|---|---|
| Bit 5 | Bit 4 | | |
| 0 | 0 | 1 character | 1 character |
| 0 | 1 | 3 characters | 6 characters |
| 1 | 0 | 9 characters | 18 characters |
| 1 | 1 | 13 characters | 26 characters |
| FCR Code | | FIFO Trigger Level of 64-byte Transmitter | FIFO Trigger Level of 128-byte Transmitter |
| Bit 5 | Bit 4 | | |
| 0 | 0 | 1 character | 1 character |
| 0 | 1 | 16 characters | 32 characters |
| 1 | 0 | 32 characters | 64 characters |
| 1 | 1 | 56 characters | 120 characters |

### 23.4.2.6 Line Control Register (LCR, Offset: 0x0C)

This register controls in a way that the transmitted characters are serialized and the received characters are assembled and checked.

**Table 23-10. Line Control Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| 7 | DLAB | R/W | Divisor Latch Access Bit (DLAB)<br>This bit must be set in order to access the DLL, DLM, and PSR registers, which program the division constants for the baud rate divider and the prescaler. |
| 6 | Set Break | R/W | This bit causes a break condition to be transmitted to the receiving UART. When it is set to logic 1, the serial output (**io_irda_sout**) is forced to the Spacing (Logic 0) state. The break is disabled by setting bits[6:0]. The Break Control bit acts only on **io_irda_sout** and has no effect on the transmitter logic, so if several characters are stored in the transmit FIFO, they will be removed from this FIFO and passed sequentially to the Transmitter Shift Register which serializes them, even if Set Break is set. This fact can be useful to establish the break time making use of the THR Empty and Transmitter Empty flags of the LSR. Firmware can follow the sequence below to assure no erroneous or extraneous characters will be transmitted because of the break:<br>Set break when transmitter is idle (LSR bit[6]).<br>Write a character with any value to THR.<br>Wait for the transmitter to become idle (LSR bit[6]), and clear break when normal transmission has to be restored. |
| 5 | Stick Parity | R/W | When bits 3, 4, and 5 are logic 1, the Parity bit is transmitted and checked as 0. If bits 3 and 5 are 1 and bit 4 is 0, then the Parity bit is transmitted and checked as 1. If bit 5 is 0, Stick Parity is disabled. Please refer to Table 23-11. |
| 4 | Even Parity | R/W | This bit is the Even Parity Select bit. When bit 3 is 1 and bit 4 is 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. Please refer to Table 23-11. |
| 3 | Parity Enable | R/W | This bit is the Parity Enable bit. When this bit is a 1, a Parity bit is generated (Transmit data) or checked (Receive data) between the last data word bit and Stop bit of the serial data. When bit 3 is 1 and bit 4 is a 1, an even number of 1s is transmitted or checked. Please refer to Table 23-11. |
| 2 | Stop Bits | R/W | This bit selects the number of stop bits to be transmitted. If cleared, only one stop bit will be transmitted. If set, two stop bits (1.5 with 5-bit data) will be transmitted before the start bit of the next character. The receiver always checks only one stop bit. Please refer to<br>Table 23-12. |
| 1 | WL1 | R/W | This bit along with WL0 defines the word length of the data being transmitted and received. Please refer to<br>Table 23-12 for the possible selections. |
| 0 | WL0 | R/W | This bit along with WL1 defines the word length of the data being transmitted and received. Please refer to<br>Table 23-12 for the possible selections. |

**Table 23-11. Parity Setting Table**

| LCR Code | | | Parity Bit (Transmitted or Checked) |
|---|---|---|---|
| Bit 5 Stick Parity | Bit 4 Even Parity | Bit 3 Parity Enable | |
| X | X | 0 | Not transmitted or checked |
| 0 | 1 | 1 | Even parity |
| 0 | 0 | 1 | Odd parity |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |

**Table 23-12. Word Length and Stop Bit Setting Table**

| LCR Code | | | Character Length (Bits) | Number of Stop Bits |
|---|---|---|---|---|
| Bit 2 Stop Bit | Bit 1 WL1 | Bit 0 WL0 | | |
| 0 | 0 | 0 | 5 | 1 |
| | 0 | 1 | 6 | |
| | 1 | 0 | 7 | |
| | 1 | 1 | 8 | |
| 1 | 0 | 0 | 5 | 1.5 |
| | 0 | 1 | 6 | 2 |
| | 1 | 0 | 7 | |
| | 1 | 1 | 8 | |

### 23.4.2.7 Modem Control Register (MCR, Offset: 0x10)

By writing this register, the user can set the modem control outputs (**io_irda_ndtr** and **io_irda_nrts**). This register also controls the loop back mode, and provides general purpose outputs.

**Table 23-13. Modem Control Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| 7 | Reserved | - | - |
| 6 | Out3 | R/W | This bit controls the general purpose, active low, output io_irda_nout3 in the same way as bit 0 controls io_irda_ndtr.<br>This bit is not found in the standard 16550 UART. |
| 5 | DMAmode2 | R/W | This bit selects the UART/SIR DMA mode.<br>The DMA mode2 affects the way in which the DMA signaling output pins (irda_nrxrdy and irda_ntxrdy) behave. |
| 4 | Loop | R/W | Loop back mode control bit. Loop back mode is intended to test the UART or SIR communication. |
| 3 | Out2 | R/W | This bit controls the general purpose, active low, output io_irda_nout2 in the same way as bit 0 controls io_irda_ndtr. |
| 2 | Out1 | R/W | This bit controls the general purpose, active low, output io_irda_nout1 in the same way as bit 0 controls io_irda_ndtr. |
| 1 | RTS (Request to Send) | R/W | This bit controls the “request to send” active low output (io_irda_nrts) in the same way as bit 0 controls io_irda_ndtr. |
| 0 | DTR (Data Terminal Ready) | R/W | This bit controls the “data terminal ready” active low output io_irda_ndtr. A 1 in this bit makes io_irda_ndtr output a 0. When this bit is cleared, io_irda_ndtr outputs a 1. |

When UART is set in the loopback mode, the following may occur:

- The serial output is connected internally to the serial input; every character sent is looped back and received.
- The input pin, **io_irda_sin**, is not used and the output pin, **io_irda_sout**, is set to ‘1’ (Inactive state).
- Four modem control inputs are internally connected to two modem control outputs plus the general-purpose outputs. This way, **io_irda_ncts** is internally controlled by **io_irda_nrts**, **io_irda_ndsr** by **io_irda_ndtr**, **io_irda_nri** by **io_irda_nout1**, and **io_irda_ndcd** by **io_irda_nout2**. That is to say; there is a non-ordered correspondence between four least significant bits of MCR and four most significant bits of MSR. Please refer to Figure 23-1 for the connections. The modem control output pins are forced to the inactive state (High).
- Four modem control input pins, **io_irda_ncts**, **io_irda_ndsr**, **io_irda_nri**, and **io_irda_ndcd**, are not used. Two modem control output pins, **io_irda_ndtr** and **io_irda_nrts**, and three user outputs, **io_irda_nout1**, **io_irda_nout2**, and **io_irda_nout3**, are set to ‘1’ (Inactive state).

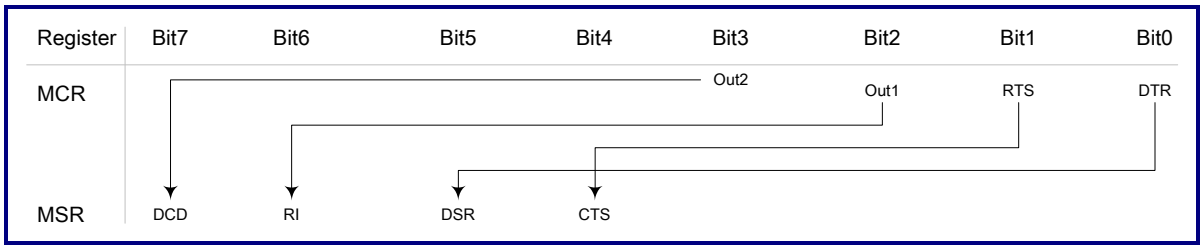



**Figure 23-2. Interconnection between MCR and MSR in Loopback Mode**

### 23.4.2.8 Line Status Register (LSR, Offset: 0x14 for Read)

This register informs the user of the status of the transmitter and the receiver. In order to get information about a received character, LSR must be read before reading that received character from RBR.

**Table 23-14. Line Status Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| 7 | FIFO Data Error | R | If FIFO is disabled (16450 mode), this bit is always zero.<br>If FIFO is active, this bit will be set as soon as any data character in the receiver FIFO has parity or framing error or the break indication active. This bit is cleared when the CPU reads the LSR and the rest of the data in the receiver FIFO do not have any of these three associated flags on. |
| 6 | Transmitter Empty | R | This bit is '1' when both THR (Or TX FIFO) and TSR (Transmitter Shift Register) are empty. Reading this bit as '1' means that no transmission is currently taking place in the io_irda_sout output pin, and that the transmission line is idle. As soon as new data is written in THR, this bit will be cleared. |
| 5 | THR Empty | R | This bit indicates that UART is ready to accept a new character for transmission. In addition, this bit causes UART to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable bit (IER [1]) is set to high.<br>In the non-FIFO mode, this bit is set whenever the 1-byte THR is empty. If THR holds data to be transmitted, this bit is immediately set when this data is passed to TSR.<br>In the FIFO mode, this bit is set when the transmitter FIFO is completely empty, being 0 if there is at least one byte in FIFO waiting to be passed to TSR for transmission. |

| Bit | Name | Type | Function |
|---|---|---|---|
| 4 | Break Interrupt | R/C | This bit is set to 1 if the receiver line input, io_irda_sin, was held at zero for a complete character time. That is to say, the positions corresponding to the start bit, the data, the parity bit (If any) and the (First) stop bit were all detected as zeroes.<br><br>Please note that a Framing Error flag always accompanies this flag. This bit is queued in the receiver FIFO in the same way as the Parity Error bit. When break occurs, only one zero character is loaded into the FIFO.<br><br>The next character transfer is enabled after io_irda_sin goes to the marking state and receives the next valid start bit.<br><br>This bit is cleared as soon as the LSR is read. |
| 3 | Framing Error | R | This bit indicates that the received character did not have a valid stop bit (i.e., a 0 was detected in the (First) stop bit position instead of a 1). This bit is queued in the receiver FIFO in the same way as the Parity Error bit. When a framing error is detected, the receiver tries to resynchronize: if the next sample is again a zero, it will be taken as the beginning of a possible new start bit.<br><br>This bit is cleared as soon as the LSR is read. |
| 2 | Parity Error | R | When this bit is set, it indicates that the parity of the received character is wrong according to the current setting in LCR. This bit is queued in the receiver FIFO, so it is associated with the particular character that had the error.<br><br>Therefore, LSR must be read before RBR: each time a character is read from RBR, the next character passes to the top of FIFO and LSR is loaded with the queued error flags corresponding to this top-of-the-FIFO character.<br><br>This bit is cleared as soon as the LSR is read. |
| 1 | Overrun Error | R | When this bit is set, a character has been completely assembled in the Receiver Shift Register without having free space to put it in the receiver FIFO or holding register. When an overrun condition appears, the result is different depending on whether the 16-byte FIFO is active or not:<br><br>When the FIFO is not active, only a 1-character Receiver Holing Register will be available. The unread data in this RBR will not be overwritten with the new character received.<br><br>When the FIFO is active, the character received in the Receiver Shift Register will be overwritten, but the data already present in the FIFO will not be changed. The Overrun Error flag will be set as soon as the overrun condition appears.<br><br>This bit is not queued in the FIFO if it is active.<br><br>This bit is cleared as soon as the LSR is read. |
| 0 | Data Ready | R | This bit is set if one or more characters have been received and are waiting in the receiver's FIFO for the user to read them. It is cleared to logic 0 by reading all of the data in the Receiver Buffer Register or the FIFO. |

### 23.4.2.9 Testing Register (TST, Offset: 0x14 for Write)

This register provides internal diagnostic capabilities if the circuit is hardware implemented.

The loop back mode is supported in the UART and SIR modes only. For FIR, because there is only one CRC module in FTUART010, FIR transmit and receive cannot be tested with the loopback mode.

**Table 23-15. Testing Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:5] | Reserved | - | - |
| 4 | TEST_CRC_ERR | W | When this bit is set, FTUART010 will generate incorrect CRC during the FIR transmission. |
| 3 | TEST_PHY_ERR | W | When this bit is set, FTUART010 will generate incorrect 4PPM encoding chips during the FIR transmission. |
| 2 | TEST_BAUDGEN | W | This bit is used to improve the baud rate generator toggle rate. |
| 1 | TEST_FRM_ERR | W | When this bit is set, FTUART010 will generate a logic 0 STOP bit during the UART transmission. |
| 0 | TEST_PAR_ERR | W | When this bit is set, FTUART010 will generate incorrect parity during the UART transmission. |

### 23.4.2.10 Modem Status Register (MSR, Offset: 0x18)

This register provides information about the status of the four modem control input pins. The four most significant bits directly provide the status of the pin, while the four least significant bits give information about changes in these pins.

Four least significant bits are used to generate an interrupt (Modem Status interrupt) if these bits are enabled by the corresponding bit in the IER. The interrupt will be generated as soon as any of these four significant bits is 1. They are reset to logic 0 whenever the Modem Status Register is read.

**Table 23-16. Modem Status Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| 7 | DCD | R | Data Carrier Detect (DCD), which is the complement of the io_irda_ndcd input. |
| 6 | RI | R | Ring Indicator (RI), which is the complement of the io_irda_nri input. |
| 5 | DSR | R | Data Set Ready (DSR), which is the complement of the io_irda_ndsr input. |
| 4 | CTS | R | Clear To Send (CTS), which is the complement of the io_irda_ncts input. |

| Bit | Name | Type | Function |
|---|---|---|---|
| 3 | Delta DCD | R | The delta-DCD flag. If set, it means that the io_irda_ndcd input has changed since the last time the microprocessor read this bit. |
| 2 | Trailing edge R1 | R | This bit is set when a trailing edge is detected in the io_irda_nri input pin; that is to say, when io_irda_nri changes from '0' to '1'. |
| 1 | Delta DSR | R | If this bit is set, it means that the io_irda_ndsr input has changed since the last time the microprocessor read this bit. |
| 0 | Delta CTS | R | If this bit is set, it means that the io_irda_ncts input has changed since the last time the microprocessor read this bit. |

#### 23.4.2.11 Scratch Pad Register (SPR, Offset: 0x1C)

**Table 23-17. Scratch Pad Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:0] | User Data | R/W | This 8-bit read/write register has no effect on the operation of the Serial Port. It is intended as a scratchpad register to be used by the programmer to hold data temporarily. |

#### 23.4.2.12 Baud Rate Divisor Latch (DL, Offset: 0x00, 0x04 when DLAB = 1)

The Divisor Latch is a 16-bit register, of which the most significant byte is held in DLM and the least significant byte is held in DLL. The division factors can be programmed from 1 to 65535. The access to these two registers, located at addresses 1 and 0 respectively, is conditioned based on the value of the DLAB bit in the LCR register. The two registers can be written and read only if this bit is 1. Otherwise, the IER, RBR and THR will be accessed instead.

These two registers (DLL and DLM), together with the Prescaler Register (PSR) select the speed at which the communication will occur. This is the baud rate at which characters will be transmitted and the expected baud rate for the characters that will be received. Only one baud rate is defined for both transmission and reception.

The baud rate is defined as the **io_irda_uclk** frequency divided by 16, divided by the contents of the PSR register, and then divided by the contents of the Divisor Latch. When DLM and DLL are programmed as 0's, there is no output clock. It is recommended to program DLL and DLM as 0's for power saving in FIR mode.

**Table 23-18. Baud Rate Divisor Latch LSB**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:0] | DLL | R/W | Baud Rate Divisor Latch Least Significant Byte |

**Table 23-19. Baud Rate Divisor Latch MSB**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:0] | DLM | R/W | Baud Rate Divisor Latch Most Significant Byte |

### 23.4.2.13 Prescaler Register (PSR, Offset: 0x08 when DLAB =1)

This 5-bit register (PSR[4:0]) adds a second programmable division factor to obtain the desired baud rate (Please refer to the description of the Divisor Latch register above). The division factor is the value hold in this register, so the maximum factor is 31 and the minimum is 0. Bits 5 to 7 are always zero. This is a non-standard register (i.e., it is not present in the industry standard 16550 UART). The input clock **io_irda_uclk** is divided by integers from 1 to 31. When PSR is a 0, there is no input clock to divisor latch unit. Therefore, programming DLL and DLM is useless when PSR is set as 0. The default value for the PSR register is 1.

This register is only accessible when the DLAB bit in LCR is set. Otherwise, the Line Status Register will be accessed.

**Table 23-20. Prescaler Register**

| Bit | Name | Type | Function |
|---|---|---|---|
| [7:5] | Reserved | - | - |
| [4:0] | PSR | R/W | Prescaler Value |

# Chapter 24

# Synchronous Serial Port Controller ($I^2S$)

This chapter contains the following sections:


- 24.1 General Description
- 24.2 Features
- 24.3 Block Diagram
- 24.4 Programming Model

## 24.1 General Description

FTSSP010 is a synchronous serial port interface that allows the host processor to serve as a master or a slave. It can connect to various devices by using serial protocol. FTSSP010 supports the Synchronous Serial Port (SSP) from Texas Instruments, the Serial Peripheral Interface (SPI) from Motorola, MICROWIRE from National Semiconductor, $I^2S$ from Philips,. The serial data formats may range from 4 bits to 32 bits in length. The SSP controller can directly use the on-chip DMA to transfer data between the external serial device and the system memory without intervention from the processor.

## 24.2 Features

- Supports the TI SSP, Motorola SPI, National Semiconductor MICROWIRE, Philips I2S
- Independent SSP clock for easy generation of bit clock
- Supports the master and slave modes
- Internally or externally controlled serial bit clock
- Internally or externally controlled frame/sync.
- Programmable frame/sync. polarity
- Programmable serial bit clock polarity, phase, and frequency
- Programmable serial bit data sequence (MSB or LSB first)
- Programmable $I^2S$ format including zero bit padding and right/left justification
- Programmable threshold interrupt of transmit/receive FIFO
- Independently programmable interrupt enable/disable
- Independently configurable transmit and receive data FIFO depth

## 24.3 Block Diagram

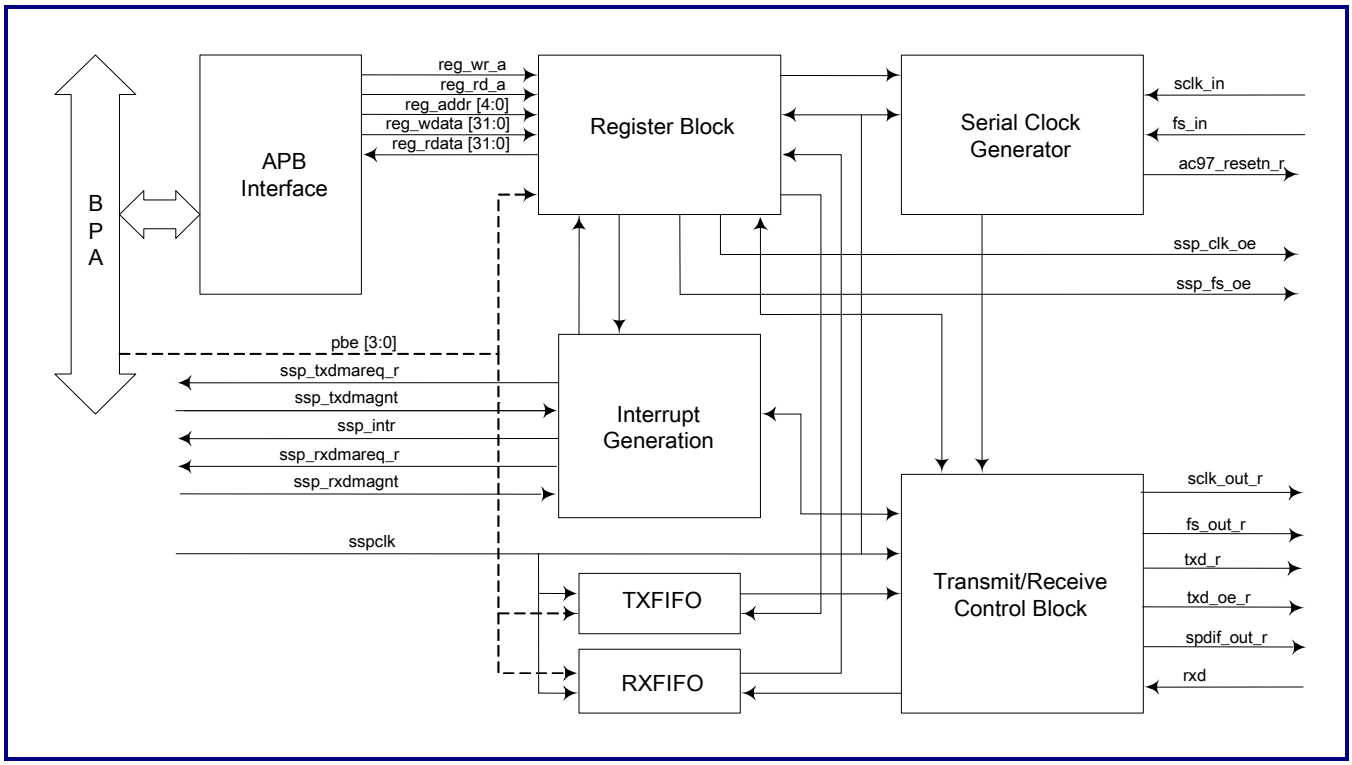



**Figure 24-1. Functional Block Diagram**

## 24.4 Programming Model

### 24.4.1 SSP Control Registers Specification

**Table 24-1. SSP Control Register Specification**

| Address | Type | Description | Reset Value |
|---|---|---|---|
| +0x00 | R/W | SSP control register 0 | 0x0000_010C |
| +0x04 | R/W | SSP control register 1 | 0x0007_8000 |
| +0x08 | R/W | SSP control register 2 | 0x0000_0002 |
| +0x0C | R | SSP status register | 0x0000_0002 |
| +0x10 | R/W | SSP interrupt control register | 0x0000_2200 |
| +0x14 | RC | SSP interrupt status register | 0x0000_0008 |
| +0x18 | R/W | SSP data register | N/A |

### 24.4.2 Register Descriptions

The following sections will provide the detailed information of the SSP control registers.

#### 24.4.2.1 SSP Control Register 0 (Offset = 0x00)

This register controls the generation and behavior of frame/sync.

**Table 24-2. SSP Control Register 0**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:15] | - | R | Reserved and read as zeros |
| [14:12] | FFMT | R/W | Frame format<br>This register defines the pre-defined frame format according to the following encoding:<br>000: Texas Instrument's Synchronous Serial Port (SSP)<br>001: Motorola's Serial Peripheral Interface (SPI)<br>010: National Semiconductor's MICROWIRE<br>011: Philips' $I^2S$<br>100: Reserved<br>101: Reserved<br>110 ~ 111: Not defined<br>If these bits are set, the SSP operation will be unpredictable. |
| 11 | - | R | Reserved and read as zeros |
| 10 | - | R | Reserved |
| [9:8] | FSDIST | R/W | Frame/Sync. and data distance<br>These bits only take effect when $I^2S$ frame format is specified. These bits define the relationship between the frame/sync and the data bits. If these bits are set to zero, the first bit of serial data will be valid in the first cycle of frame/sync. If non-zero is specified, FSDIST defines the number of SCLK cycles between first bit of valid received data and frame start. For the standard $I^2S$ frame format, these bits must be set to 1. |
| 7 | LBM | R/W | Loop-back mode<br>If this bit is set to '1,' the transmitted data will be connected to the received data internally. This is used for self test only. If this bit is set to '0,' SSP operates in the normal mode and the transmitted/received data are independent. |
| 6 | LSB | R/W | Bit sequence indicator<br>If this bit is set to '0,' the Most Significant Bit (MSB) of a data word will be transmitted or received first. If this bit is set to '1,' the Least Significant Bit (LSB) of a data word will be transmitted or received first. This bit only takes effect when the $I^2S$ frame format is specified. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 5 | FSPO | R/W | Frame/Sync polarity<br>If this bit is set to '0,' the frame/sync will be treated as active high. If this bit is set to '1,' the frame/sync will be treated as active low. This bit only takes effect when the $I^2S$ or MWR frame format is specified. |
| 4 | FSJSTFY | R/W | Data justify<br>This bit is valid only when the $I^2S$ frame format is specified. If this bit is set to '0' and the Padding Data Length (PDL) in the control bit is not zero, the number of zeros specified in PDL will be appended in the back of the serial data. If this bit is set to '1,' the padding data will be in front of the serial data. |
| [3:2] | OPM | R/W | Operation mode<br>If the SSP, SPI, or MICROWIRE frame format is specified, these bits specify the operation modes as follows:<br>00, 01: Slave mode<br>10, 11: Master mode<br><br>If the $I^2S$ frame format is specified, these bits define the operation modes as follows:<br>11: Master stereo mode<br>10: Master mono mode<br>01: Slave stereo mode<br>00: Slave mono mode |
| 1 | SCLKPO | R/W | SCLK polarity<br>This bit only takes effect if Motorola's SPI is specified.<br>When this bit is set to '0', SCLK will remain low when SSP is idle.<br>When this bit is set to '1', SCLK will remain high when SSP is idle. |
| 0 | SCLKPH | R/W | SCLK phase<br>This bit only takes effect if Motorola's SPI is specified. This bit defines the relationship between SCLK and frame/sync.<br>When this bit is set to '0,' SCLK will start toggling after one SCLK cycle time when frame/sync is activated.<br>When this bit is set to '1,' SCLK will start running after half an SCLK cycle time when frame/sync is activated. |

### 24.4.2.2 SSP Control Register 1 (Offset = 0x04)

This register defines the clock divider and the data length of transfer. Table 24-3 shows the detailed format of SSP control register 1.

**Table 24-3. SSP Control Register 1**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | PDL | R/W | Padding data length |
| [23:21] | | R | Reserved and read as zeros |
| [20:16] | SDL | R/W | Serial data length<br>These bits define the bit length of a transmitted/received data word. The actual data length equals to these bits plus 1. |
| [15:0] | SCLKDIV | R/W | SCLK divider<br>These bits define the serial clock divider. |

When the $I^2S$ frame format is specified, PDL will define the bit length of the padding bits in front/back of the data word. If PDL is set to '0,' no padding bit will be inserted. If the value specified is a non-zero value, the actual number of zeroes will be inserted/appended in front/back of the data word.

If the National Semiconductor MICROWIRE frame format is specified, the PDL defines the bit length of the first phase (Transmitted in the master mode and received in the slave mode), and the actual data length is equal to this register plus 1. The maximum value of this register should not exceed 31 if National Semiconductor's MICROWIRE is specified. The minimum value of this register should be more than 1.

The minimum value of SDL should not be less than four, and the maximum value of this register refers to the width of the configured FIFO.

The frequency of the internally generated serial clock is controlled by SCLKDIV[15:0]. The frequency of the internally generated serial clock is determined by the following formula:

$$f_{SCLK} = \frac{f_{SSPCLK}}{2 \times (SCLKDIV + 1)}$$

Note: The $f_{SSPCLK}$ is SSP0_CLK/SSP1_CLK/SSP2_CLK. Please refer to the Figure 4-1.

### 24.4.2.3 SSP Control Register 2 (Offset = 0x08)

**Table 24-4. SSP Control Register 2**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:7] | - | R | Reserved and read as zeros |
| 6 | SSPRST | R/W | SSP reset<br>The software reset of the SSP controller state machine<br>Writing a '1' to this bit will result in the reset of the SSP controller.<br>Writing a '0' has no effect. |
| 5 | - | - | Reserved |
| 4 | - | - | Reserved |
| 3 | TXFCLR | W | Transmit FIFO clear<br>This is a write only bit.<br>Writing this bit to '1' will clear all data in the transmit FIFO.<br>Writing this bit to '0' takes no effect.<br>Reading this bit always gets a value of zero. |
| 2 | RXFCLR | W | Receive FIFO clear<br>This is a write only bit.<br>Writing this bit to '1' will clear all data in the receive FIFO.<br>Writing this bit to '0' takes no effect.<br>Reading this bit always gets a value of zero. |
| 1 | TXDOE | R/W | Transmit data output enable<br>This bit is only valid when the SSP slave mode is specified. In the multi-slave system, it is possible for an SSP master to broadcast a message to all slaves within the system while ensuring that only one slave drives data onto its serial output line. In such a system, the rxd line from the multiple slaves could be tied together. To operate in such system, TXDOE may be cleared if the SSP slave is not supposed to drive the txd_r line. When TXDOE is set, the SSP will drive data on the transmit data line; when TDOE is not set, the SSP will not drive data on the transmit data line. If the $I^2S$ frame format is specified, this bit defines either the receiving (Recording) mode (0) or the simultaneous transmitting/receiving (Playing/Recording) mode (1). |
| 0 | SSPEN | R/W | The SSP enable<br>If this bit is set to '1,' the SSP controller will start transmitting/receiving data if possible. If this bit is set to '0,' the serial data will stop toggling. Please note that when the SSPEN bit is cleared in an active operation, SSP will stop transmitting or receiving after the current word had been transmitted or received. |

Writing a '1' to SSPRST will reset the SSP state machine even if the normal transmission or reception is in progress. Usually, this bit is written when the software stops the SSP controller by setting SSPEN to '0.' For the slave mode of the SSP controller, the serial clock is controlled by the external master and will be stopped under certain circumstances, and caused the SSP state machine to halt. Writing a '1' to this bit puts the SSP controller back to the idle state when carrying out the following operations.

#### 24.4.2.4 SSP Status Register (Offset = 0x0C)

**Table 24-5. SSP Status Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:17] | - | R | Reserved and read as zeros |
| [16:12] | TFVE | R | Transmit FIFO valid entries<br>These bits indicate the number of entries in the transmit FIFO waiting to be transmitted. |
| [11:9] | - | R | Reserved and read as zero |
| [8:4] | RFVE | R | Receive FIFO valid entries<br>These bits indicate the number of entries in the receive FIFO waiting for DMA or host processor to read them. |
| 3 | - | R | Reserved and read as zero |
| 2 | BUSY | R | Busy indicator<br>If this bit is read as '1,' it indicates the SSP is transmitting and/or receiving data. If this bit is read as '0,' it indicates SSP is idle or disabled. |
| 1 | TFNF | R | Transmit FIFO not full<br>This bit is set to '1' whenever the transmit FIFO is available for DMA or host processor to write. TFNF is cleared to '0' when FIFO is completely full. |
| 0 | RFF | R | Receive FIFO full<br>This bit is set to '1' whenever the receive FIFO is full. RFF is cleared to 0 when the DMA controller or host processor reads data from the receive FIFO. |

### 24.4.2.5 Interrupt Control Register (Offset = 0x10)

**Table 24-6. Interrupt Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | R | Reserved and read as zeros |
| [15:12] | TFTHOD | R/W | Transmit FIFO threshold<br>If the valid data in transmit FIFO is equal to or less than the actual threshold, the DMA request and/or the interrupt will be asserted. If this bit is set to '0,' the interrupt will occur when the transmit FIFO is completely shifted out. |
| [11:8] | RFTHOD | R/W | Receive FIFO threshold<br>If the valid data in the receive FIFO is equal to or greater than the actual threshold, the DMA request and/or the interrupt will be asserted. If this bit is set to 0, the interrupt will be disabled. |
| 7 | - | R | Reserved and read as zero |
| 6 | - | - | Reserved |
| 5 | TFDMAEN | R/W | Transmit DMA request enable<br>If this bit is set to '1,' the DMA request will be issued when the transmit FIFO threshold (TFLETH is set to '1') is hit. If this bit is set to '0,' no DMA request will be issued. |
| 4 | RFDMAEN | R/W | Receive DMA request enable<br>If this bit is set to '1,' the DMA request will be issued when the receive FIFO threshold (RFGETH is set to '1') is hit. If this bit is set to '0,' no DMA request will be issued. |
| 3 | TFTHIEN | R/W | Transmit FIFO threshold interrupt enable<br>If this bit is set to '1,' the interrupt will be issued when the valid entries in the transmit FIFO are less than or equal to the threshold value. |
| 2 | RFTHIEN | R/W | Receive FIFO threshold interrupt enable<br>If this bit is set to '1,' the interrupt will be issued when the valid entries in the receive FIFO are greater than or equal to threshold value. |
| 1 | TFURIEN | R/W | Transmit FIFO underrun interrupt enable<br>If this bit is set to '1,' the transmit FIFO underrun will cause the SSP to assert interrupt. If this bit is set to '0,' the interrupt will be masked even when the transmit FIFO underrun happens. |
| 0 | RFORIEN | R/W | Receive FIFO overrun interrupt enable<br>If this bit is set to '1,' the receive FIFO overrun will cause the SSP to assert interrupt. If this bit is set to '0,' the interrupt will be masked even when the receive FIFO overrun occurs. |

### 24.4.2.6 Interrupt Status Register (Offset = 0x14)

**Table 24-7. Interrupt Status Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:5] | - | R | Reserved and read as zero |
| 4 | - | - | Reserved |
| 3 | TFTHI | R/RC | Transmit FIFO threshold interrupt<br>If TFTHOD is set to non-zero, this bit is set to '1' when the transmit FIFO is equal to or less than the threshold. If the valid entries in the transmit FIFO are larger than TFTHOD, this bit will be cleared automatically.<br>If TFTHOD is set to '0,' this bit is set to '1' when the last data in the transmit FIFO is completely shifted out. This bit will never be cleared to '0' until user reads this bit. |
| 2 | RFTHI | R | Receive FIFO threshold interrupt<br>If the receive FIFO is equal to or greater than the threshold, this bit will be set to '1.'<br>This bit will be cleared when the condition above is removed. |
| 1 | TFURI | RC | Transmit FIFO underrun interrupt<br>If the transmit logic tries to retrieve data from empty transmit FIFO, this bit will be set to '1.'<br>This bit will never be cleared to '0' until users read this bit. |
| 0 | RFORI | RC | Receive FIFO overrun interrupt<br>If the receive logic tries to receive data when the receive FIFO is full, this bit will be set to '1.'<br>This bit will never be cleared to '0' until users read this bit. |

### 24.4.2.7 SSP Transmit/Receive Data Register (Offset = 0x18)

The SSP controller separates transmit and receive FIFOs, which are 32-bit wide and can be configured separately (Please refer to Chapter 4 for more details). The transmitted and received data occupy the same address space. The write operation writes data into the transmit FIFO and the read operation reads data from the receive FIFO. If the size of a serial data length is less than 32 bits, data will be right-justified automatically at the time of reception or transmission.

# Chapter 25

# Secure Digital Memory Host Controller (SDC)

This chapter contains the following sections:


- 25.1 General Description
- 25.2 Features
- 25.3 Block Diagram
- 25.4 Programming Model

## 25.1 General Description

The host controller of the Secure Digital (SD) memory card functions as the master in an SD memory card interface. It controls the communication between the AHB/APB bus and the SD card. The core supports the SD bus of the SD operations and the MMC bus of the MMC operation as well.

## 25.2 Features

- Supports SD memory card protocol, version 2.0
  - Does not include the SPI mode
- Supports SDIO bus protocol, version 1.10
  - Supports multi-function and combo cards
  - Supports 1-bit or 4-bit SD data transfer mode with the SDIO mode
  - Supports multi-block transfer interrupt
  - Supports read wait operation with SDIO
- Supports MMC bus protocol, version 4.1
  - Does not include the SPI mode
  - Supports three data bus width modes: 1-bit, 4-bit, and 8-bit
  - Supports interrupt mode
- Supports AMBA 2.0 bus
  - Hardware configurable AHB or APB
  - Supports data access in byte, half-word, and word in the AHB interface
- Supports DMA for large data transfers
- Configurable 16-word/8-word/4-word data FIFO depths
- Supports built-in generation and checks for 7-bit and16-bit CRC data
- Variable clock rate: 0 MHz ~ 50 MHz for the SD card, 0 MHz ~ 52 MHz for the MMC card
- Supports hot insertion/removal
- Supports write-protect for the SD card

## 25.3 Block Diagram

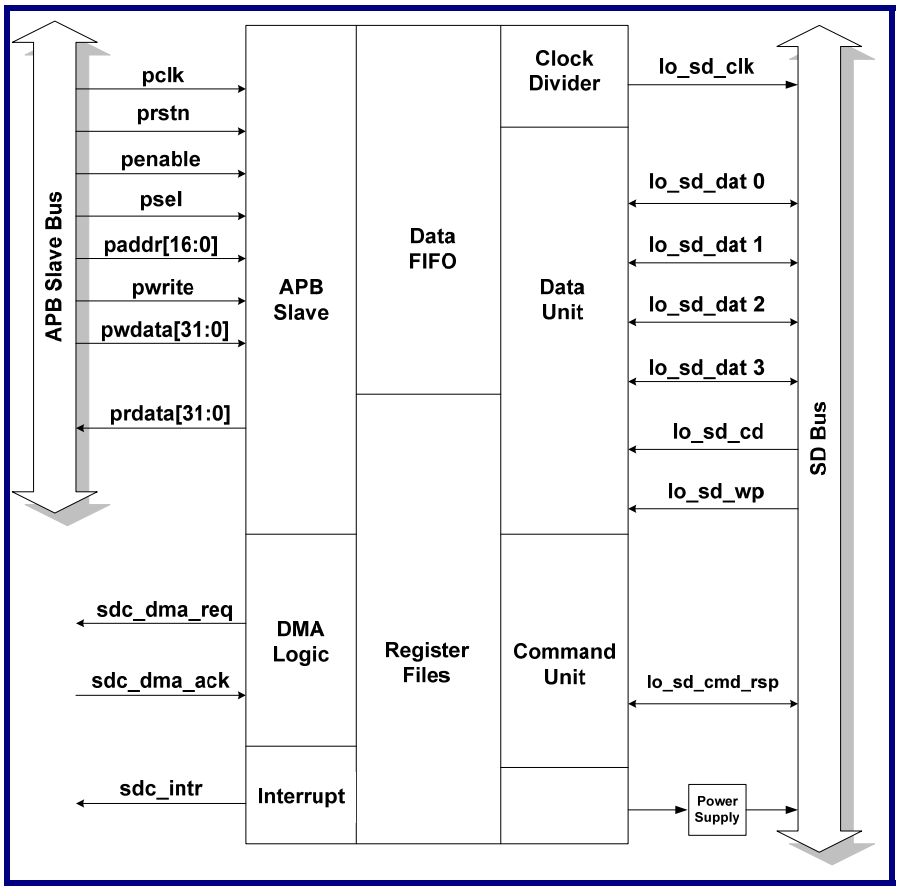



**Figure 25-1. Block Diagram**

The main building blocks of the SD host controller are the APB slave ports, which can be hardware-configured to the AHB slave port, SD master, CRC logic, power management, and clock management.

## 25.4 Programming Model

### 25.4.1 Summary of SD Host Controller Registers

**Table 25-1. Summary of SD Host Controller Registers**

| Offset | Type | Width | Description | Reset Value |
|---|---|---|---|---|
| +0x00 | R/W | 11 | Command register | 0000_0000 |
| +0x04 | R/W | 32 | Argument register | 0000_0000 |
| +0x08 | R | 32 | Response register0 | 0000_0000 |
| +0x0C | R | 32 | Response register1 | 0000_0000 |
| +0x10 | R | 32 | Response register2 | 0000_0000 |
| +0x14 | R | 32 | Response register3 | 0000_0000 |
| +0x18 | R | 7 | Responded command register | 0000_0000 |
| +0x1C | R/W | 11 | Data control register | 0000_0000 |
| +0x20 | R/W | 32 | Data timer register | 0000_0000 |
| +0x24 | R/W | 32 | Data length register | 0000_0000 |
| +0x28 | R | 17 | Status register | 0000_0000 |
| +0x2C | W | 17 | Clear register | 0000_0000 |
| +0x30 | R/W | 17 | Interrupt mask register | 0000_0000 |
| +0x34 | R/W | 5 | Power control register | 0000_0010 |
| +0x38 | R/W | 9 | Clock control register | 0000_01ff |
| +0x3C | R/W | 6 | Bus width register F[6] | 0000_0009 or 0000_0001 |
| +0x40 | R/W | 32 | Data window register | - |
| +0x44 | R/W | 8 | MMC interrupt response time register | 0000_0042 |
| +0x48 | R/W | 4 | General-purpose output | 0000_0000 |
| +0x6C | R/W | 32 | SDIO control register1 | 0000_0000 |
| +0x70 | R/W | 2 | SDIO control regiset2 | 0000_0000 |
| +0x74 | R | 24 | SDIO status register | 0000_0000 |
| +0x9C | R | 9 | Feature register | - |
| +0xA0 | R | 32 | Revision register | - |
| +0x1000 | R/W | 32 | Virtual memory register | - |

[6] Please refer to Section 29.4.2.13 for more details.

### 25.4.2 Register Description

The following sections describe the SD host controller registers in detail.

#### 25.4.2.1 Command Register

The command register contains the command index and relevant information about this command. Table 25-2 shows the bit assignment of the command register. Before setting a new command to the command register, the device driver should make sure that the last command is finished or has responded. When SDC_RSTn is set, each register will be reset to its default value, except for the response registers 0 ~ 3. The time division between two sequential SDC_RSTn signals must be longer than two SD clock cycles.

Bit[9] of this register will be automatically cleared by hardware once the internal command state machine is activated.

Bit[10] of this register will be automatically cleared by hardware once the internal command state machine is activated and is cleared by RSP_CRC_OK in the status register.

**Table 25-2. Command Register (Offset: 0x00)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:12] | Reserved | - | - |
| 11 | MMC_INT_STOP | R/W | When this bit is set, the MMC host controller will send out R5 response to inform MMC card to exit the Wait-IRQ state. Before setting this bit, user should confirm that the CMD_EN signal is '0'. |
| 10 | SDC_RST | R/W | When this bit is set, the SD host controller will be reset. |
| 9 | CMD_EN | R/W | When this bit is set, the command will be enabled. |
| 8 | APP_CMD | R/W | This bit indicates that the command is an application-specific command. |
| 7 | LONG_RSP | R/W | When this bit is set, this command will wait for the 136-bit long response. |
| 6 | NEED_RSP | R/W | When this bit is set, this command will wait for a response from the card. |
| [5:0] | CMD_IDX | R/W | The command index is sent as part of a command. |

### 25.4.2.2 Argument Register

This register contains a 32-bit argument that is sent to a card as part of a command. Table 25-3 shows the bit assignment of the argument register.

**Table 25-3. Argument Register (Offset: 0x04)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | CMD_ARG | R/W | Argument of a command |

### 25.4.2.3 Response Registers 0 ~ 3

A response may be 48-bit (R1, R3, and R6) or 136-bit (R2) wide, depending on the command type. The card status, which is part of the response, may be 32-bit or 127-bit wide. The card status reported from the card will be saved in these registers. Table 25-4 shows the bit assignment of the response registers.

**Table 25-4. Response 0-3 Register (Offset: 0x08 ~ 0x14)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | CARD_STA | R | Card status sent by card |

In Table 25-5, the most significant bit of the card status is received first. For long responses, the least significant bit, RESPONSE0, is always set to '1'.

**Table 25-5. Short Responses and Long Responses (Offset: 0x08 ~ 0x14)**

| Register Name | Short Response | Long Response |
|---|---|---|
| RESPONSE0 | Card status[31:0] | Card status[31:0] |
| RESPONSE1 | Unused | Card status[63:32] |
| RESPONSE2 | Unused | Card status[95:64] |
| RESPONSE3 | Unused | Card status[127:96] |

### 25.4.2.4 Responded Command Register

This register contains the index of the last command that receives a response. Table 26-6 shows the bit assignment of the responded command register. RSP_CMD_IDX is only meaningful for R1-type and R6-type responses.

**Table 25-6. Responded Command Register (Offset: 0x18)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:7] | Reserved | - | - |
| 6 | RSP_CMD_APP | R | This bit indicates that the command is an application-specific command. |
| [5:0] | RSP_CMD_IDX | R | Response command index |

### 25.4.2.5 Data Control Register

Table 25-7 shows the bit assignment of the data control register. A counter loads the value from the BLK_SIZE bit and starts to decrease as the data is being transferred. This counter decreases to zero when the block of data completes transfer. The SD host controller does not support the multiple-block read or write when the block size is 1 byte or 2 bytes. To read or write the 1-byte or 2-byte data, the single block read or write command should be used. The device driver should not set to a block size that the card does not support.

Bit[6] of this register will be automatically cleared by the hardware once the internal data state machine is activated.

When bit[5] is set to logic 1, the DMA request signal, sdc_dma_req, is asserted if the data transfer is not finished. When the data transfer is complete, the sdc_dma_req signal is de-asserted. Setting bit[5] to logic 0 will de-assert the sdc_dma_req signal immediately.

**Table 25-7. Bit Assignment of Data Control Register (Offset: 0x1c)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:11] | Reserved | - | - |
| 10 | FIFO_RST | W | FIFO reset<br>Write '1' to this bit will reset the counter of FIFO. |

| Bit | Name | Type | Description |
|---|---|---|---|
| [9:8] | DMA_TYPE | R/W | DMA type<br>This bit indicates which type of DMA transaction is used. These two bits are valid when the DMA_EN signal is enabled.<br>00: Single read/write<br>01: Burst four reads/writes<br>10: Burst eight reads/writes<br>11: Revised |
| 7 | FIFOTH | R/W | FIFO threshold<br>0: FIFO_EMPTY or FIFO_FULL in the status register will be asserted when the FIFO is empty or full.<br>1: FIFO_URUN or FIFO_ORUN in the status register will be asserted when the FIFO is half-empty or half-full. |
| 6 | DATA_EN | R/W | Enable data transfer cycle |
| 5 | DMA_EN | R/W | 0: Disable the DMA transfer<br>1: Enable the DMA transfer |
| 4 | DATA_WRITE | R/W | 0: Read data from the card<br>1: Write data to the card |
| [3:0] | BLK_SIZE | R/W | Data size per block (Bytes) |

**Table 25-8. Data Block Size**

| BLK_SIZE | Block Length (Byte) |
|---|---|
| [15:12] | Reserved |
| 11 | $2^{11}$ = 2048 |
| 10 | $2^{10}$ = 1024 |
| …. | - |
| 3 | $2^3$ = 8 |
| 2 | $2^2$ = 4 |
| 1 | $2^1$ = 2 |
| 0 | $2^0$ = 1 |

### 25.4.2.6 Data Timer Register

**Table 25-9. Data Timer Register (Offset: 0x20)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | DATA_TIMER | R/W | Data timeout period (In-card bus clock period) |

### 25.4.2.7 Data Length Register

This register contains the number of data bytes to be transferred. Table 25-10 shows the bit assignment of the data length register. To start a data transfer, the data must be written to the data timer register and data length register before being written to the data control register. The value in the data length register must be multiples of the block size.

**Table 25-10. Data Length Register (Offset: 0x24)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | DATA_LEN | R/W | Data bytes to be transferred<br>The maximum volume cannot exceed 128 MB in the SDIO and combo card applications. |

### 25.4.2.8 Status Register

Table 25-11 shows the bit assignments of the read-only status register. This register can be cleared by writing '1' to the corresponding clear register.

- The DATA0_STATUS flag responds to the status of the primary input pin, io_sd_data0_in.
- The SDIO_IRPT flag is asserted when the SDIO host receives an interrupt of the SDIO device.
- The FIFO_ORUN flag can only be asserted in the read cycle, which means that the read data is available (FIFO is half-full or entirely full, depending on the FIFO threshold) in the data FIFO. If the data length decreases to zero, the CPU will read the data output.
- The FIFO_URUN flag can only be asserted in the write cycle, which means that FIFO is entirely empty or half-empty, depending on the FIFO threshold. The CPU should send write data to FIFO only when this bit is polled as '1'.

- When a command does not send out a response, only the CMD_SENT bit should be set. When a command sends out a response, either the RSP_CRC_OK bit or the RSP_CRC_FAIL bit will be set.
- When a data block is transferred, the DATA_CRC_OK bit or the DATA_CRC_FAIL bit will be set. The DATA_END bit will be set when the data transfer is finished and FIFO is empty.
- When the WRITE_PROTECT bit is set to '1', the device driver prevents all write commands from being sent to the SD card.
- Whenever the card is removed or inserted, the CARD_CHANGE bit will be set to '1' (An interrupt will occur if the corresponding mask bit is set to '1'), and the device driver should poll the CARD_DETECT bit to determine whether the card was inserted or removed.
- It is the responsibility of the device driver to prevent all write commands from being issued when the WRITE_PROT bit (Bit 12) is set to '1'.
- The WRITE_RPOT and CARD_DETECT bits directly provide the pin status .

The interrupt logic generates the interrupt request signal, sdc_intr, when at least one of the selected status flags (Except WRITE_PROT and CARD_DETECT) is set to high. A status flag will not generate the interrupt request if the corresponding mask flag is low.

**Table 25-11. Status Register (Offset: 0x28)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:18] | Reserved | - | - |
| 17 | DATA0_STATUS | R | Real status of the pin data0<br>The interrupt will be asserted when the data0 pin is high. |
| 16 | SDIO_IRPT | R | SDIO card interrupt |
| [15:13] | Reserved | - | - |
| 12 | WRITE_PROT | R | 0: Card is not write-protected.<br>1: Card is write-protected. |
| 11 | CARD_DETECT | R | 0: Card is inserted.<br>1: Card is removed. |
| 10 | CARD_CHANGE | R | Card is inserted or removed. |
| 9 | FIFO_ORUN | R | Data FIFO is available for users to read. |
| 8 | FIFO_URUN | R | Data FIFO is available for users to write. |
| 7 | DATA_END | R | Data transfer is complete. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 6 | CMD_SENT | R | Command sent (No response is required) |
| 5 | DATA_CRC_OK | R | Data block is sent/received and CRC check is passed. |
| 4 | RSP_CRC_OK | R | Command response is received and CRC check is passed. |
| 3 | DATA_TIMEOUT | R | Data read/programming timeout |
| 2 | RSP_TIMEOUT | R | Command response timeout |
| 1 | DATA_CRC_FAIL | R | Data block is sent/received but the CRC check is failed. |
| 0 | RSP_CRC_FAIL | R | Command response is received but the CRC check is failed. |

#### 25.4.2.9 Clear Register

The clear register is a write-only register. Writing '1' to the corresponding bits can clear the corresponding status register. Table 25-12 shows the bit assignments of the clear register.

**Table 25-12. Clear Register (Offset: 0x2c)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:17] | Reserved | - | - |
| 16 | CLR_SDIO_IRPT | W | Clear the SDIO interrupt flag |
| [15:11] | Reserved | - | - |
| 10 | CLR_CARD_CHANGE | W | Clear the CARD_CHANGE flag |
| 9 | Reserved | - | - |
| 8 | Reserved | - | - |
| 7 | CLR_DATA_END | W | Clear the DATA_END flag |
| 6 | CLR_CMD_SENT | W | Clear the CMD_SENT flag |
| 5 | CLR_DATA_OK | W | Clear the DATA_CRC_OK flag |
| 4 | CLR_RSP_OK | W | Clear the RSP_CRC_OK flag |
| 3 | CLR_DATA_TIMEOUT | W | Clear the DATA_TIMEOUT flag |
| 2 | CLR_RSP_TIMEOUT | W | Clear the RSP_TIMEOUT flag |
| 1 | CLR_DATA_FAIL | W | Clear the DATA_CRC_FAIL flag |
| 0 | CLR_RSP_FAIL | W | Clear the RSP_CRC_FAIL flag |

### 25.4.2.10 Interrupt Mask Register

Table 25-13 shows the bit assignments of the interrupt mask register. This register determines which status flag generates an interrupt request by setting the corresponding bit to '1'.

**Table 25-13 Interrupt Mask Register (Offset: 0x30)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:18] | Reserved | - | - |
| 17 | Mask17 | R/W | Mask the DATA0_STATUS flag |
| 16 | Mask16 | R/W | Mask the SDIO interrupt flag |
| 15 | Mask15 | R/W | Mask the PLAIN_TEXT_READY flag |
| 14 | Mask14 | R/W | Mask the CP_BUF_READY flag |
| 13 | Mask13 | R/W | Mask the CP_READY flag |
| [12:11] | Reserved | - | - |
| 10 | Mask10 | R/W | Mask the CARD_CHANGE flag |
| 9 | Mask9 | R/W | Mask the FIFO_ORUN flag |
| 8 | Mask8 | R/W | Mask the FIFO_NRUN flag |
| 7 | Mask7 | R/W | Mask the DATA_END flag |
| 6 | Mask6 | R/W | Mask the CMD_SENT flag |
| 5 | Mask5 | R/W | Mask the DATA_CRC_OK flag |
| 4 | Mask4 | R/W | Mask the RSP_CRC_OK flag |
| 3 | Mask3 | R/W | Mask the DATA_TIMEOUT flag |
| 2 | Mask2 | R/W | Mask the RSP_TIMEOUT flag |
| 1 | Mask1 | R/W | Mask the DATA_CRC_FAIL flag |
| 0 | Mask0 | R/W | Mask the RSP_CRC_FAIL flag |

### 25.4.2.11 Power Control Register

The power control register controls the external power supply. The supported voltage of the memory card is stored in the OCR register. Since the operating voltage can be any value of between 2.0 V and 3.6 V, an appropriate output voltage can be adjusted by this register. Table 25-14 shows the bit assignments of the power control register.

**Table 25-14. Power Control Register (Offset: 0x34)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:5] | Reserved | - | - |
| 4 | SD_POWER_ON | R/W | Control the external power supply<br>1: Power on<br>0: Power off |
| [3:0] | SD_POWER | R/W | Control the external power supply output range |

### 25.4.2.12 Clock Control Register

In the card identification mode, the maximum clock frequency is 400 kHz ($f_{OD}$). During the data transfer, the maximum clock frequency rate ($f_{PP}$) can not exceed 50 MHz for the SD memory card, and 52 MHz for the MMC card. After the Relative Card Address (RCA) is published from the card, the clock frequency can be switched from $f_{OD}$ to $f_{PP}$. Users should avoid changing the clock frequency rate or disabling the clock when the command or data is still in progress to prevent any unpredictable result. The termination of the memory clock should not violate the requirement of the memory card specification. Table 25-15 shows the bit assignments of the clock control register. After the card initialization process, bit 7 of this register should also be programmed to show the type of card detected. This bit is used to synchronize the output signal from the memory card. When the SD card is in the normal speed mode (25 MHz), bit 9 will be set to '0' to handle normal speed transfer. After the SD card is switched to the high speed mode (50 MHz), the bit 9 needs to be set to '1' to handle high speed transfer. The default value of bit 9 is set to '0'.

**Table 25-15. Clock Control Register (Offset: 0x38)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:10] | Reserved | - | - |
| 9 | HI_SPEED | R/W | 1: High speed mode transfer<br>0: Normal speed mode transfer |
| 8 | CLK_DIS | R/W | 1: SD_CLK is disabled.<br>0: SD_CLK is enabled. |
| 7 | CLK_SD | R/W | 1: SD memory card<br>0: MMC memory card |
| [6:0] | CLK_DIV | R/W | Control the clock frequency of io_sd_clk<br>io_sd_clk = SD_CLK/2x (CLK_DIV + 1) |

### 25.4.2.13 Bus Width Register

Bit[5] of this register is used by the host to detect the plugging or removing a card. The bus width supported by an SD memory card is defined in the SCR register. Bits[4:0] show the bit assignment of the bus width register. The values of bit[4] and bit[3] are determined by the synthesis option. When bit[3] is read as logic 1, it indicates that this controller is implemented with a 4-bit wide memory data bus. When bit[4] is read as logic 1, it indicates that this controller is implemented with an 8-bit wide memory data bus. Writing any value to these bits will be meaningless. Bits[2:0] can be programmed by software to transfer data by using a 8-bit, 4-bit, or 1-bit data bus.

**Table 25-16. Bus Width Register (Offset: 0x3c)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:6] | Reserved | - | - |
| 5 | CARD_DETECT | R/W | 1: Card detection through DATA3<br>0: Card detection through the switch of the connector |
| [4:3] | WIDE_BUS_SUPPORT | R | 10: 8-bit wide bus is implemented.<br>01: 4-bit wide bus is implemented.<br>00: 1-bit wide bus is implemented. |
| 2 | WIDE_4_BUS | R/W | Bus width = 4 (DAT[0:3]) when the WIDE_BUS_SUPPORT bit is enabled. |
| 1 | WIDE_8_BUS | R/W | Bus width = 8 (DAT[0:7]) when the WIDE_BUS_SUPPORT bit is enabled. |
| 0 | SINGLE_BUS | R/W | Bus width = 1 (DAT0) |

### 25.4.2.14 Data Window Register

This register acts as a data port when an AHB/APB device accesses the SD memory card. Whenever the address of an AHB/APB cycle matches the address of the data window register, it is forwarded as a data read/write cycle. For byte and half-word accesses, the data window register can be accessed with the sequential or continuous manner. The data order is based on the little-endian format. For example, when bits[7:0] are accessed, the next access will start from bits[15:8]. In case of a write operation, the data are transferred into the FIFO when the register is 32-bit wide. In case of a read operation, the 32-bit data in FIFO is transferred to the data window register. And the data window register can be accessed in byte, half-word, or word.

**Table 25-17. Data Window Register (Offset: 0x40)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | DATA_WIN | R/W | Data access port |

### 25.4.2.15 MMC Interrupt Response Time

This register indicates that the MMC host will receive the R5 response after the MMC host sent CMD40 to the MMC card. The rsp_timeout interrupt will be asserted if the host does not receive a response within the assigned time.

**Table 25-18. MMC Interrupt Response Time (Offset: 0x44)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [7:0] | MMC_INT_TIME | R/W | MMC interrupt response time |

### 25.4.2.16 General-purpose Output Register

This register drives directly to the output signal of FTSDC010.

**Table 25-19. General-purpose Output Register (Offset: 0x48)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:4] | Reserved | - | - |
| [3:0] | sdc_gpo | R/W | Direct drive the levels of four general output ports |

### 25.4.2.17 SDIO Control Register1

This register indicates the control part of the SDIO host.

**Table 25-20. SDIO Control Register (Offset: 0x6c)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:15] | SDIO_BLK_NO | W/R | The total block number for a data transfer<br>This field should be set before a data transfer. The maximum block number in a single data transfer is 131071; the minimum block number is 1. |
| 14 | SDIO_EN | W/R | Set this bit to enable the SDIO related function. |

| Bit | Name | Type | Description |
|---|---|---|---|
| 13 | READ_WAIT_EN | W/R | Set this bit when the SDIO card supports the Read_Wait feature |
| 12 | SDIO_BLK_MODE | W/R | Set this bit when the host wants to initiate a multiple block transaction |
| [11:0] | SDIO_BLK_SIZE | W/R | Data size (Bytes) per block<br>In a single block transaction, the maximum block size is 512.<br>In a multiple block transaction, the maximum block size is 2048. Once SDIO_EN is set, this field records the data block size no matter it is a memory or an I/O data transfer. |

### 25.4.2.18 SDIO Control Register2

This register indicates the control part of the SDIO host.

**Table 25-21. SDIO Control Register (Offset: 0x70)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:2] | Reserved | - | - |
| 1 | Susp_cmd_abort | W/R | To abort a suspended transaction, this bit must be set first before the abort command can be sent. |
| 0 | Susp_read_wait | W/R | This bit must be set before sending out the suspend command. The bus will keep the read-wait state until this bit is cleared. |

### 25.4.2.19 SDIO Status Register

This register indicates the status of the SDIO host.

**Table 25-22. SDIO Status Register (Offset: 0x74)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | Reserved | - | - |
| [23:17] | FIFO_REMAIN_NO | R | This field records the number of valid data (Words) remained in the FIFO. |
| [16:0] | SDIO_BLK_CNT | R | This field records the number of blocks remained in a data transfer. |

### 25.4.3 Virtual Memory Register

This register is not a physically implemented register. Rather, it is an address at which the DMA can read or write in byte/half-word access. DMA can access continuous byte/half-word/word data through the virtual address. The virtual address should be accessed in a sequential or continuous manner. The maximum data access size is limited to 64k bytes. If the access size reaches the boundary of 64k bytes boundary, the access address must be re-started from the start address (0x10000) to avoid of exceeding the register offset boundary.

**Table 25-23. Virtual Memory Register (Offset: 0X10000)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | Virtual_Mem_Addr | W/R | For continuous and sequential byte/half-word/word DMA access |

# Chapter 26

# I²C Bus Interface Controller

This chapter contains the following sections:


- 26.1 General Description
- 26.2 Features
- 26.3 Block Diagram
- 26.4 Programming Model

## 26.1 General Description

The $I^2C$ bus interface controller, FTIIC010, allows the host processor to serve as a master or slave in the $I^2C$ bus. Data are transmitted to and received from the $I^2C$ bus via a buffered interface.

## 26.2 Features

- Supports standard and fast modes by programming the clock division register
- Supports 7-bit, 10-bit, and general-call addressing modes
- Glitch suppression throughout the debounce circuit
- Programmable slave address
- Supports master-transmit, master-receive, slave-transmit, and slave-receive modes
- Supports multi-master mode
- General-call address detection in slave mode
- Not supports START byte procedure
- Supports system manager bus 2.0 in master and slave modes
- Supports all system manager bus 2.0 protocol commands except quick command and host notify protocol

## 26.3 Block Diagram

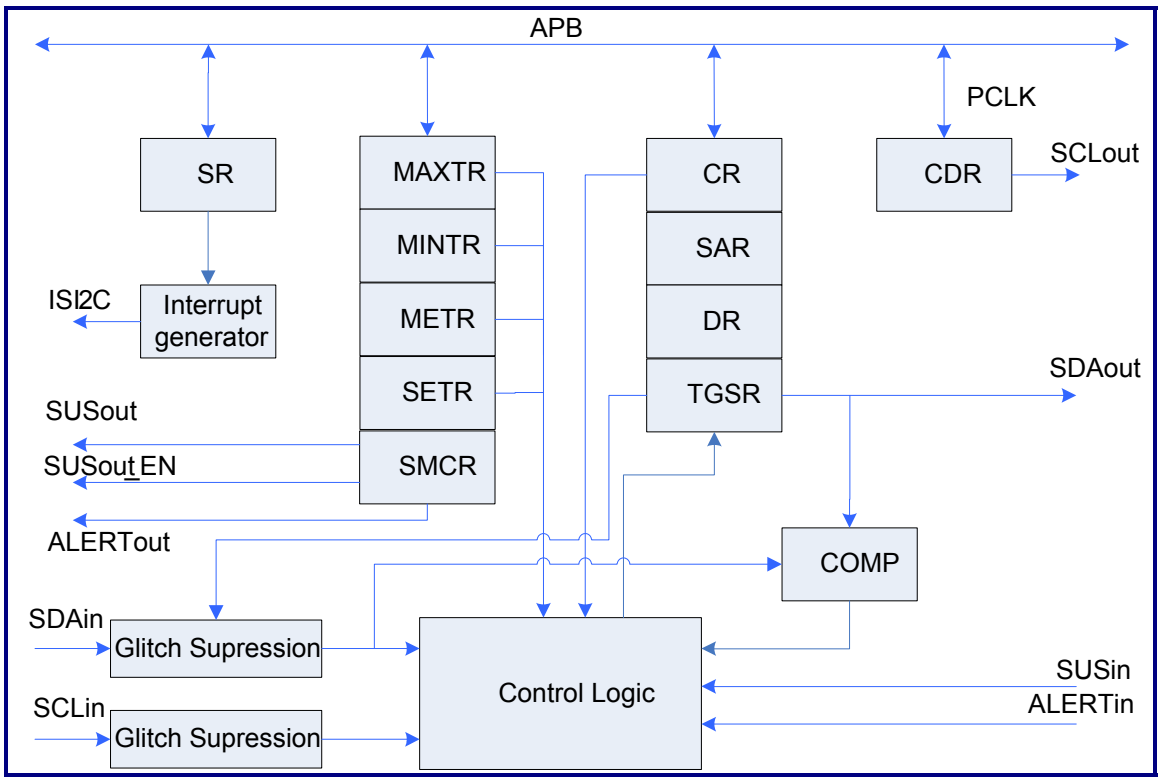



**Figure 26-1. Functional Block Diagram**

The main building blocks of the $I^2C$ bus interface controller include the register files, control logic, SCLout generator, and debounce circuit.

## 26.4 Programming Model

### 26.4.1 Summary of FTIIC010 $I^2C$ Controller Registers

The following registers are associated with the $I^2C$ bus interface controller. These registers are allocated within the peripheral memory-mapped addresses of the host processor.

**Table 26-1. Summary of FTIIC010 I²C Controller Register**

| Offset Address | Type | Description | Reset Value |
|---|---|---|---|
| 0x00 | R/W | I²C Control Register (CR) | 0x0000_0000 |
| 0x04 | R/RC | I²C Status Register (SR) | 0x0000_0000 |
| 0x08 | R/W | I²C Clock Division Register (CDR) | 0x0000_0000 |
| 0x0C | R/W | I²C Data Register (DR) | 0x0000_0000 |
| 0x10 | R/W | I²C Slave Address Register (SAR) | 0x0000_0000 |
| 0x14 | R/W | I²C Setup/Hold Time and Glitch Suppression Setting Register (TGSR) | 0x0000_0401 |
| 0x18 | R | I²C Bus Monitor Register (BMR) | 0x0000_0003 |
| 0x1C | R/W | SM Bus Control Register (SMCR) | 0x0000_0000 |
| 0x20 | R/W | Maximum Timeout Register (MAXTR) | 0x003f_ffff |
| 0x24 | R/W | Minimum Timeout Register (MINTR) | 0x003f_ffff |
| 0x28 | R/W | Master Extend Time Register (METR) | 0x000f_ffff |
| 0x2C | R/W | Slave Extend Time Register (SETR) | 0x001f_ffff |
| 0x30 | R | I²C Revision Register | - |
| 0x34 | R | I²C Feature Register | - |

### 26.4.2 Register Descriptions

The following sections describe the details of the I²C controller register.

#### 26.4.2.1 I²C Control Register (CR) (Offset: 0x00)

The host processor uses the Control Register (CR) to control the I²C interface controller to transmit and receive data from I²C bus.

Please note that SCL_EN must be set to '1' to enter the master mode unless user wants to release SCLout in a special case. Set SCL_EN to '0' to enter the slave mode. The START and STOP bits can only be set in the master mode.

Table 26-2 shows the bit assignment of the control register.

**Table 26-2. $I^2C$ Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:18] | - | - | Reserved |
| 17 | Test_bit | R/W | Special test mode; this bit must be set to 0. |
| 16 | SDA_LOW | R/W | If set, SDAout is tied to 0. For a normal case, it is suggested to set this bit to 0. |
| 15 | SCL_LOW | R/W | If set, the SCLout is tied to 0. For a normal case, it is suggested to set this bit to 0. |
| 14 | STARTI_EN | R/W | If set, this bit enables the $I^2C$ controller to interrupt the host processor when FTIIC010 $I^2C$ controller detects a start condition happening on the $I^2C$ bus. |
| 13 | ALI_EN | R/W | If set, this bit enables the $I^2C$ controller to interrupt the host processor when the $I^2C$ controller loses the arbitration in the master mode. |
| 12 | SAMI_EN | R/W | If set, this bit enables the $I^2C$ controller to interrupt the host processor when the $I^2C$ controller detects a slave address that matches the SAR register or a general call address (when GC_EN is set). |
| 11 | STOPI_EN | R/W | If set, this bit enables the $I^2C$ controller to interrupt the host processor when the $I^2C$ controller detects a stop condition happening on the $I^2C$ bus. |
| 10 | BERRI_EN | R/W | If set, this bit enables the $I^2C$ controller to interrupt the host processor when the $I^2C$ controller detects non-ACK responses from the slave device after one byte of data has been sent in the master mode. |
| 9 | DRI_EN | R/W | If set, this bit enables the $I^2C$ controller to interrupt the host processor when the $I^2C$ controller DR register has received one data byte from the $I^2C$ bus. |
| 8 | DTI_EN | R/W | If set, this bit enables the $I^2C$ controller to interrupt the host processor when the $I^2C$ controller DR register has transmitted one data byte onto the $I^2C$ bus. |
| 7 | TB_EN | R/W | When Transfer Byte Enable (TB_EN) is set, the $I^2C$ controller is ready to receive or transmit one byte. Otherwise, the $I^2C$ controller will insert the wait state by pulling SCLout low in the $I^2C$ bus.<br>Please note that the TB_EN will be automatically cleared when the device receive or transmit one byte or detect the stop/start condition. |
| 6 | ACK/NACK | R/W | The acknowledge signal sent by the $I^2C$ controller when the $I^2C$ controller is in master-receive or slave-receive mode<br>0: ACK<br>1: NACK |
| 5 | STOP | R/W | The $I^2C$ controller initiates a stop condition after transferring the next data byte on the $I^2C$ bus when $I^2C$ is in the master mode. |
| 4 | START | R/W | The $I^2C$ controller initiates a start condition when $I^2C$ bus is idle, or initiates a repeated start condition after transferring the next data byte on the $I^2C$ bus in the master mode. |
| 3 | GC_EN | R/W | Enable the $I^2C$ controller to respond to a general call message as a slave |
| 2 | SCL_EN | R/W | Enable the $I^2C$ controller clock output for the master mode operation |
| 1 | I2C_EN | R/W | Enable the $I^2C$ bus interface controller |

| Bit | Name | Type | Description |
|---|---|---|---|
| 0 | I2C_RST | R/W | Reset the I$^2$C controller<br>This bit will be automatically cleared after two PCLK clocks. |

### 26.4.2.2 I$^2$C Status Register (SR) (Offset: 0x04)

The I$^2$C and SMBus interrupts are signaled through the ISI2C pin. The ISI2C pin is set when both the interrupt enable bits (CR) and (SMCR) in the control register and the corresponding status register are set. When the I$^2$C controller interrupt is asserted, software reads the SR bits to check the status of the I$^2$C controller.

SR is also used to clear the following interrupts by reading the register status:

- DR receive data is completed
- DR transmit data is completed
- Slave address is detected
- Bus error is detected
- Start condition is detected
- Stop condition is detected
- Arbitration loss is detected
- Leave suspend mode
- Receive alert response address
- Enter suspend mode
- Alert active detected
- Detect maximum timeout
- Detect minimum timeout
- Detect master extend time
- Detect slave extend time

Table 26-3 shows the bit assignment of the status register.

**Table 26-3. I²C Status Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:22] | - | - | Reserved |
| 21 | SAL | RC | Set when SM bus controller loses arbitration when operating in the slave mode |
| 20 | DDA | RC | Set when slave receive address is device default address |
| 19 | ARA | RC | Set when slave receive address is alert response address |
| 18 | RESUME | RC | Set when the system leaves the suspend mode |
| 17 | SUSPEND | RC | Set when the system enters the suspend mode |
| 16 | ALERT | RC | Set when the master needs to read the Alert Response Address (ARA) |
| 15 | TOUTMax | RC | Set when single clock held low longer than TTIMEOUT,MAX |
| 14 | TOUTMin | RC | Set when single clock held low longer than TTIMEOUT,MIN |
| 13 | MEXT | RC | Set when cumulative clock low extend time longer than TLow_MEXT |
| 12 | SEXT | RC | Set when cumulative clock low extend time longer than TLow_SEXT |
| 11 | START | RC[7] | Set when the I²C controller detects a start condition on the I²C bus |
| 10 | AL | RC | Set when the I²C controller loses arbitration when operating in the master mode. |
| 9 | GC | RC | Set when the I²C controller receives a slave address that matches the general call address, when the I²C controller is operating in the slave mode |
| 8 | SAM | RC | Set when the I²C controller receives a slave address that matches the address in the slave register (SAR) when the I²C controller is operating in the slave mode |
| 7 | STOP | RC | Set when the I²C controller detects a stop condition in the I²C bus. |
| 6 | BERR | RC | Set when the I²C controller detects non-ACK responses from the slave device after one byte of data has been transmitted when the I²C controller is operating in the master mode |
| 5 | DR | RC | Set when the data register (DR) received one new data byte from the I²C bus |
| 4 | DT | RC | Set when the data register (DR) transmitted one data byte to the I²C bus |
| 3 | BB | R | Set when the I²C bus is busy, but the I²C controller is not involved in the transaction |
| 2 | I²CB | R | Set when the I²C controller is busy, i.e. during the time period between the START and STOP |
| 1 | ACK | R | Set when the I²C controller receives or sends non-acknowledgements |
| 0 | RW | R | Set when the I²C controller serves in a master-receive or slave-transmit mode. |

[7] RC means "Read and clear."

### 26.4.2.3 I²C Clock Division Register (CDR) (Offset: 0x08)

The I²C clock division register (CDR) defines the divided value used to generate the I²C SCL clock. This register is used with an internal 18-bit counter. When the SCL enable bit in the control register is set, this counter decrements from the programmable value to zero, and then reloads the programmable value and decrements again. Each time the counter reaches zero, the SCL line transaction starts from high to low or vice versa, depending on the current state. This register can be configured to select the transfer speed needed in the I²C bus.

Please note that because the controller supports the I²C bus speed up to 400 kHz, the minimum PCLK frequency must be 6 MHz.

**Table 26-4. I²C Clock Division Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:18] | - | - | Reserved |
| [17:0] | COUNT | R/W | Counter value used to generate an I²C clock (SCLout) from the internal bus clock PCLK. The relation between PCLK and I²C bus clock (SCLout) is shown in the following equation, where GSR is TGSR[12:10]<br>SCLout = PCLK/(2*(COUNT + 2) + GSR) |

### 26.4.2.4 I²C Data Register (DR) (Offset: 0x0c)

The I²C Data Register (DR) is used by the host processor to transmit and receive data from the I²C bus. The DR is accessed by either the host processor or the I²C controller Shift Register (SHR). Data coming from the I²C bus interface are received by the DR register after a full data byte has been received. Data going out of the I²C bus interface are written into the DR register by the host processor and sent to the serial bus.

When the I²C controller is in the transmit mode, the host processor writes data into the DR register over the APB bus. This occurs when a master transition is initiated or when a data transmit interrupt is signaled. Data are moved from the DR register to the SHR register when the transfer byte enable (TB_EN) in the control register is set. The data transmit interrupt is signaled when one byte of data has been transferred on the I²C bus. If DR is not written by the host processor before the next byte package, the I²C controller will insert a wait state until the host processor writes to the DR and sets the Transfer Byte Enable bit. When the I²C controller is in the received mode, the processor reads the DR register data over the APB bus,

when the received interrupt of DR is signaled. When the I$^2$C controller has received one new data byte, it will automatically clear transfer byte enable bit (TB_EN) and issue the ACK/NACK on the I$^2$C bus. After issuing ACK/NACK, the I$^2$C controller will insert DR received interrupt to processor. Users must set the Transfer Byte bit again for the next byte transfer on the I$^2$C bus.

**Table 26-5. I$^2$C Data Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:8] | - | - | Reserved |
| [7:0] | DR | R/W | Buffer for I$^2$C bus data transmission and reception |

### 26.4.2.5 I$^2$C Slave Address Register (SAR) (Offset: 0x10)

The I$^2$C slave address register defines the I$^2$C controller 10-bit or slave address to which the processor responds when the I$^2$C controller operates in the slave mode. The host processor writes this register before enabling I$^2$C operation. The register is fully programmable so it can be set to a value other than the fixed slave peripheral address preexisted in the system.

**Table 26-6. I$^2$C Slave Address Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 31 | EN10 | R/W | 10-bit addressing mode enable bit |
| [30:10] | - | - | Reserved |
| [9:7] | SAR | R/W | The most significant 3-bit address to which the I$^2$C controller responds when I$^2$C operates in the 10-bit addressing slave mode (EN10 = 1). When EN10 = 0, the I$^2$C controller ignores these three bits. |
| [6:0] | SAR | R/W | The 7-bit address to which the I$^2$C controller responds when the I$^2$C operates in the 7-bit addressing slave mode (EN10 = 0) or the least significant 7-bit address to which the I$^2$C controller responds when the I$^2$C operates in the 10-bit addressing slave mode. |

### 26.4.2.6 I²C Set/Hold Time and Glitch Suppression Setting Register (TGSR) (Offset: 0x14)

The I²C set/hold time and glitch suppression setting register (TGSR[9:0]) defines the PCLK clock cycles. After the SCL bus goes low, the data will be sent to the SDA bus when the I²C controller serves as a transmitter, or an acknowledgement will be sent to the SDA bus when the I²C controller serves as a receiver.

The I²C set/hold time and glitch suppression setting register (TGSR[12:10]) defines the values of the PCLK clock period when the I²C bus interface has a built-in glitch suppression logic. Glitches are suppressed according to TGSR[12:10] * PCLK clock period. For example, with a 66-MHz (15 ns period) PCLK clock and TGSR[12:10] = 3'b100, glitches of 60 ns or less are suppressed. With a 40-MHz (25 ns period) clock and TGSR[12:10] = 2'b010, glitches of 50 ns or less are suppressed. This is within the 50 ns glitch suppression specification. The only limitation is: CDR > 3 + GSR + TSR.

**Table 26-7. I²C Set/Hold Time and Glitch Suppression Setting Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:2] | - | - | Reserved |
| [12:10] | GSR | R/W | These bits define the values of PCLK clock period when the I²C Bus Interface has built-in glitch suppression logic. Glitch is suppressed according to "GSR * PCLK" clock period. |
| [9:0] | TSR | R/W | These bits define the delay values of the PCLK clock cycles that the data or acknowledgement will be driven into the I²C SDA bus after the I²C SCL bus goes low. The actual delay value is GSR+TSR+4. Figure 26-2 shows the relationship.<br>Note: TSR cannot be set to zero. |

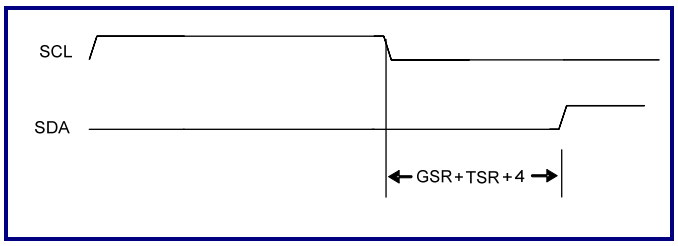



**Figure 26-2. Relationship among TSR, SCL, and SDA**

### 26.4.2.7 I²C Bus Monitor Register (BMR) (Offset: 0x18)

**Table 26-8. I²C Bus Monitor Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:2] | - | - | Reserved |
| 1 | SCLin | R | This bit continuously reflects the value of the SCLin pin. |
| 0 | SDAin | R | This bit continuously reflects the value of the SDAin pin. |

### 26.4.2.8 I²C Revision Register (Offset: 0x30)

**Table 26-9. I²C Revision Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | REVISION | R | These bits represent the revision numbers. |

### 26.4.2.9 I²C Feature Register (Offset: 0x34)

**Table 26-10. I²C Feature Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:1] | - | - | Reserved |
| 0 | SMBUS | R | 1: SM bus is configured.<br>0: No SM bus is configured. |

### 26.4.2.10 SM Control Register (SMCR) (Offset: 0x1C)

The host processor uses the SM Control Register (SMCR) to control the SMBus suspend and alert outputs. Because suspend is an optional signal, some devices can not detect if the system is in the suspend mode or not. To prevent these devices from transferring data in the system suspend mode, the slave device should pull down SCL and do not release SCL until SMBUS resumes when the device issues start in suspend mode. The I²C bus interface controller can only support this operation in the slave mode.

**Table 26-11. SM Control Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:12] | - | - | Reserved |
| 11 | SUSOUT_EN | R/W | If set, enable the suspend output. It is only used in the master mode. |
| 10 | SUS_LOW | R/W | If set, the system will issue suspend after stop. |
| 9 | ALERT_LOW | R/W | If set, the system will pull down ALERTout. |
| 8 | SAL_EN | R/W | If set, enable the SM bus controller to interrupt the host processor when the slave arbitration loses. |
| [7:6] | - | - | Reserved |
| 5 | RSM_EN | R/W | If set, enable the SM bus controller to interrupt the host processor when the system leaves the suspend mode. |
| 4 | SUS_EN | R/W | If set, enable the SM bus controller to interrupt the host processor when the system enters the suspend mode. |
| 3 | ALERT_EN | R/W | If set, enable the SM bus controller to interrupt the host processor when the system needs to read the Alert Response Address (ARA). |
| 2 | TOUT_EN | R/W | If set, enable the SM bus controller to interrupt the host processor when the clock held low single is longer than TTIMEOUT_MAX or TTIMEOUT_MIN. |
| 1 | MEXT_EN | R/W | If set, enable the SM bus controller to interrupt the host processor when the cumulative clock low extend time is longer than TLow_MEXT. |
| 0 | SEXT_EN | R/W | If set, enable the SM bus controller to interrupt the host processor when the cumulative clock low extend time is longer than TLow_SEXT. |

### 26.4.2.11 SM Maximum Timeout Register (MAXTR) (Offset: 0x20)

This register can be configured to set the maximum timeout on the SM bus. Please note that PCLK will be used as the clock of the Timer.

**Table 26-12. SM Maximum Timeout Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:22] | - | - | Reserved |
| [21:0] | TMAX | R/W | The register defines the count for maximum timeout |

### 26.4.2.12 SM Minimum Timeout Register (MINTR) (Offset: 0x24)

**Table 26-13. SM Minimum Timeout Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:22] | - | - | Reserved |
| [21:0] | TMIN | R/W | The register defines the count for minimum timeout |

### 26.4.2.13 SM Master Extend Time Register (METR) (Offset: 0x28)

**Table 26-14. SM Master Extend Time Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:20] | - | - | Reserved |
| [19:0] | TMEXT | R/W | The register defines the count for master access time from "START to ACK," "ACK to ACK," and "ACK to STOP." |

### 26.4.2.14 SM Slave Extend Time Register (SETR) (Offset: 0x2C)

**Table 26-15. SM Slave Extend Time Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:21] | - | - | Reserved |
| [20:0] | TSEXT | R/W | The register defines the count for slave access time from “START to STOP.” |

# Chapter 27

# GPIO Controller

This chapter contains the following sections:


- 27.1 General Description
- 27.2 Features
- 27.3 Block Diagram
- 27.4 Programming Model

## 27.1 General Description

FTGPIO010 is a user-programmable general-purpose input/output controller. It is used to input or output data from the system and device. Each GPIO can be programmed as an input or output.

GPIO can also be an interrupt input that is supported at the rising edge, falling edge, both edge, and the high/low level interrupt sense types.

GPIO provides up to 32 programmable I/O ports, and each port can be independently programmed.

## 27.2 Features

- Up to 32 independent input, output, and output enable buses for bidirectional I/O pins
- Each port can be bypassed
- Each port can individually trigger the GPIO interrupt when the INTR option is set and programmed as the input pin
- Triggers interrupt generation of each port at rising edge, falling edge, both edges, or at high level or low level when the INTR option is set
- Each port can be pulled high or pulled low when the "Pull" option is set (Programming I/O pad required)
- Each port can choose the pre-scaled or PCLK clock source when the INTR option is set
- Separately sets or clears output data bit
- Sets all ports as inputs by hardware reset

## 27.3 Block Diagram

Figure 27-1 shows the block diagram of the Faraday GPIO controller.

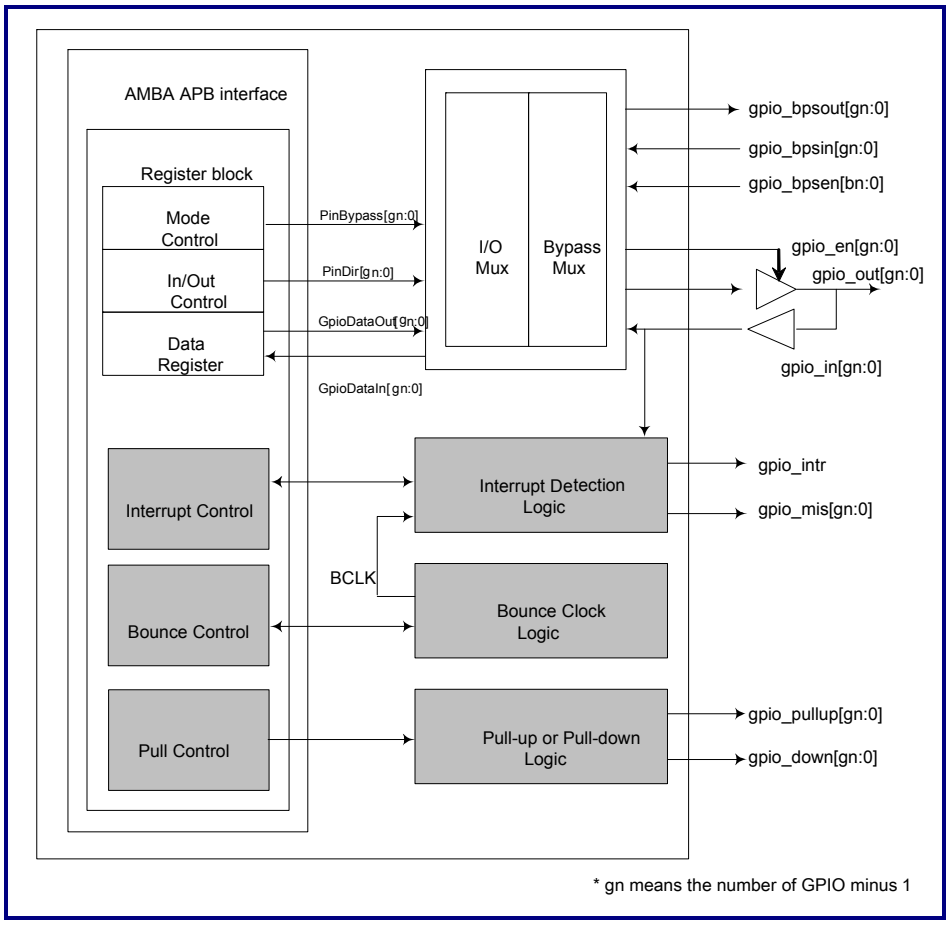



**Figure 27-1. Block Diagram**

GPIO supports the data input/output, data bit set/clear, data bypass mode, variable interrupt detection, "Pull" function, and bounce clock. The blocks in grey can be removed by using hardware configuration. The pin number of the Faraday GPIO can also be set according to the requirement of the user. The allowable range of the pin number is from 1 to 32.

## 27.4 Programming Model

### 27.4.1 Summary of General Purpose I/O Registers

Table 27-1 shows the offset, type, width, reset value, name, and configuration option of each GPIO programming register.

**Table 27-1. FTGPIO010 Register Summary**

| Offset | Type | Width | Reset | Name | Configuration | Description |
|---|---|---|---|---|---|---|
| 0x00 | R/W | [gn:0] | 0x0 | GpioDataOut | None | GPIO data output register |
| 0x04 | R | [gn:0] | 0x0 | GpioDataIn | None | GPIO data input register |
| 0x08 | R/W | [gn:0] | 0x0 | PinDir | None | GPIO direction register<br>0: Input<br>1: Output |
| 0x0C | R/W | [gn:0] | 0x0 | PinBypass | None | GPIO bypass register<br>0: No bypass<br>1: Bypass |
| 0x10 | W | [gn:0] | 0x0 | GpioDataSet | None | GPIO data bit set register<br>When writing to this register, the corresponding bits in the data register are set to 1, and the other bits remain unchanged. |
| 0x14 | W | [gn:0] | 0x0 | GpioDataClear | None | GPIO data bit clear register<br>When writing to this register, the corresponding bits in the data register are cleared to 0, and the other bits remain unchanged. |
| 0x18 | R/W | [gn:0] | 0x0 | PinPullEnable | PULL | GPIO pull-up register<br>0: Pin is not pulled.<br>1: Pin is pulled. |
| 0x1C | R/W | [gn:0] | 0x0 | PinPullType | PULL | GPIO pull-high/pull-low register<br>0: Pin is pulled low.<br>1: Pin is pulled high. |
| 0x20 | R/W | [gn:0] | 0x0 | IntrEnable | INTR | GPIO interrupt enable register<br>0: Pin interrupt is disabled.<br>1: Pin interrupt is enabled. |
| 0x24 | R | [gn:0] | 0x0 | IntrRawState | INTR | GPIO interrupt raw status register<br>0: Interrupt is not detected.<br>1: Interrupt is detected. |

| Offset | Type | Width | Reset | Name | Configuration | Description |
|---|---|---|---|---|---|---|
| 0x28 | R | [gn:0] | 0x0 | IntrMaskedState | INTR | GPIO interrupt masked status register<br>0: Interrupt is not detected or masked.<br>1: Interrupt is detected and not masked. |
| 0x2C | R/W | [gn:0] | 0x0 | IntrMask | INTR | GPIO interrupt mask register<br>0: Mask is disabled.<br>1: Mask is enabled. |
| 0x30 | W | [gn:0] | 0x0 | IntrClear | INTR | GPIO interrupt clear<br>0: No effect<br>1: Clear interrupt |
| 0x34 | R/W | [gn:0] | 0x0 | IntrTrigger | INTR | GPIO interrupt trigger method register<br>0: Edge trigger<br>1: Level trigger |
| 0x38 | R/W | [gn:0] | 0x0 | IntrBoth | INTR | GPIO edge-trigger interrupt by single or both edges<br>0: Single edge<br>1: Both edges |
| 0x3C | R/W | [gn:0] | 0x0 | IntrRiseNeg | INTR | GPIO interrupt triggered at the rising or falling edge<br>0: Rising edge<br>1: Falling edge<br>GPIO interrupt triggered by high or low level<br>0: High level<br>1: Low level |
| 0x40 | R/W | [gn:0] | 0x0 | BounceEnable | INTR | GPIO pre-scale clock enable<br>When enabled, the PCLK will be divided by the BouncePreScale clocks. This signal is used to extend the clock cycle of detecting interrupt.<br>0: Disable<br>1: Enable |
| 0x44 | R/W | [23:0] | 0x7D0 | BouncePreScale | INTR | GPIO Pre-scale, used to adjust different PCLK frequencies<br>The allowable range is from 0x1 to 0xFFFFFF. |
| 0x7C | R | [31:0] | 0x-- | RevisionNum | None | GPIO revision number |

### 27.4.2 Register Descriptions

The following sections provide the details of the general-purpose I/O registers.

#### 27.4.2.1 GpioDataOut

The GpioDataOut register is the GPIO data out register. When PinDir indicates the pin is an output, the GpioDataOut register is connected to io_out. When PinDir indicates that the pin is an input, GpioDataOut can hold the data.

**Table 27-2. GPIO Data-Out Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | GpioDataOut | Read/Write | None |

Note: GpioDataOut[I] is dedicated for io_out[I]. I is ranging from 0 to gn.

#### 27.4.2.2 GpioDataIn

The GpioDataIn register is the GPIO data in register. When the PinDir indicates that the pin is an input, the GpioDataIn will latch gpio_in at PCLK rising edge. When PinDir indicates that the pin is an output, the GpioDataIn register is a “don't care” register.

**Table 27-3. GPIO Data In Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | GpioDataIn | Read | None |

Note: GpioDataIn[I] is dedicated for io_out[I]. I is ranging from 0 to gn.

### 27.4.2.3 PinDir

The PinDir register controls gpio_en. When PinDir indicates that the pin is an output, the related bits of gpio_en are set to 1. Otherwise, the related bits of gpio_en are set to 0.

**Table 27-4. Pin Direction Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | PinDir | Read/Write | None |

Note: PinDir[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.4 PinBypass

The PinBypass register controls the bypass mode. When the PinBypass register indicates that the pin is in the bypass mode, gpio_en is connected to gpio_bps_en, gpio_in is connected to gpio_bps_out, and gpio_bps_in is connected to gpio_out.

**Table 27-5. Pin Bypass Mode Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | PinBypass | Read/Write | None |

Note: PinBypass[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.5 GpioDataSet

GpioDataSet is the bit operation logic. When writing to this address, if some bits of GpioDataSet are '1,' the related bits of GpioDataOut will be set. For example, if GpioDataOut[7:0] = 0x23, and pwdata[7:0] from the APB is 0x47, when writing to the GpioDataSet address, the result of GpioDataOut[7:0] will be 0x67.

**Table 27-6. GPIO Data Bit Set Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | GpioDataSet | Write | None |

### 27.4.2.6 GpioDataClear

GpioDataClear is the bit operation logic. When writing to this address, if some bits of GpioDataClear are 1s, the related bits of GpioDataOut will be cleared. For example, if GpioDataOut[7:0] = 0x23, and pwdata[7:0] from the APB is 0x47, when writing to the GpioDataClear address, the result of GpioDataOut[7:0] will be 0x20.

**Table 27-7. GPIO Data Bit Clear Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | GpioDataClear | Write | None |

### 27.4.2.7 PinPullEnable

The PinPullEnable register controls gpio_pullup and gpio_down. If PinPullEnable indicates that the pin is disabled to Pull, gpio_pullup and gpio_down are masked to 0.

**Table 27-8. Pin Pull Enable Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | PinPullEnable | Read/Write | Require Pull option |

Note: PinPullEnable[I] is dedicated for PAD[I]. I is ranging from 0 to gn.

### 27.4.2.8 PinPullType

PinPullType controls gpio_pullup and gpio_down. If PinPullEnable is enabled and PinPullType is set to 0, gpio_pullup will be set to 0, and gpio_down will be set to 1. Please refer to the pull truth table in Table 27-10.

**Table 27-9. Pin Pull Type Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | PinPullType | Read/Write | Require Pull option |

Notes:

1. PinPullType[I] is dedicated for PAD[I]. I is ranging from 0 to gn.
2. The pull function requires that the I/O pad can be pulled.

**Table 27-10. Pull Truth Table**

| PinPullEnable | PinPullType | gpio_pullup | gpio_down |
|---|---|---|---|
| 0 | X | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

### 27.4.2.9 IntrEnable

The IntrEnable register controls the enable or disable interrupt detection logic. It is a mask of the interrupt detection logic. When the pin direction is the input and the interrupt detection is enabled, the pin can accept interrupt from pad. The sensed state is stored in the IntrMaskedState register (Masked by IntrEnable). Before turning on IntrEnable, the programmer can clear the masked state by writing a 1 to IntrClear to ensure the initial state.

**Table 27-11. Interrupt Enable Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrEnable | Read/Write | Require INTR option |

Note: IntrEnable[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.10 IntrRawState

The IntrRawState register is the raw result of the interrupt detection. When IntrEnable is enabled, the IntrRawState register will reflect the interrupt detection status. The programmer can poll this register to detect an interrupt.

**Table 27-12. Interrupt Raw State Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrRawState | Read | Require INTR option |

Note: IntrRawState[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.11 IntrMaskedState

The IntrMaskedState register is the masked result of the interrupt detection. The IntrMaskedState register is controlled by IntrEnable, IntrRawState, and IntrMask registers.

**Table 27-13. Interrupt Masked State Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrMaskedState | Read | Require INTR option |

Note: IntrMaskedState[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.12 IntrMask

The IntrMask register is the mask register of the interrupt detection. It masks the IntrRawState register. For example, if IntrEnable[0] = 1, IntrRawState[0] = 1, and IntrMask[0] = 1, then IntrMaskedState[0] will never change to 1.

**Table 27-14. Interrupt Mask Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrMask | Read/Write | Require INTR option |

Note: IntrMask[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.13 IntrClear

IntrClear is the bit operation logic. If some bits of pwdata are set, when we write to the IntrClear address, the related bit of IntrMaskedState and IntrRawState will be cleared.

**Table 27-15. Interrupt Clear Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrClear | Write | Require INTR option |

### 27.4.2.14 IntrTrigger

The IntrTrigger register indicates the interrupt trigger method of each pin. If IntrTrigger is 0, the interrupt is triggered at the edge; otherwise, the interrupt is triggered at the level.

**Table 27-16. Interrupt Trigger Method Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrTrigger | Read/Write | Require INTR option |

Note: IntrTrigger[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.15 IntrBoth

The IntrBoth register indicates that the edge trigger is by both edges or single edge. If IntrTrigger indicates that the edge trigger and IntrBoth is 0, the interrupt edge trigger is done by the single edge. If IntrTrigger indicates that the edge trigger and IntrBoth is 1, the interrupt edge trigger is done by both edges.

**Table 27-17. Interrupt Both Edge Trigger Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrBoth | Read/Write | Require INTR option |

Note: IntrBoth[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.16 IntrRiseNeg

The IntrRiseNeg register indicates whether the edge trigger is at the rising edge or falling edge. If IntrTrigger is the edge trigger, IntrBoth is the single edge, and IntrRiseNeg is '0,' the interrupt edge trigger is done at the rising edge. If IntrTrigger is the edge trigger, IntrBoth is the single edge, and IntrRiseNeg is '1,' the interrupt edge trigger is done at the falling edge.

**Table 27-18. Interrupt Rising or Falling Edge Trigger Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | IntrRiseNeg | Read/Write | Require INTR option |

Note: IntrRiseNeg[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.17 BounceEnable

The BounceEnable register controls the bounce function. If BounceEnable is on, the interrupt detection is sampled by the extended clock. The extension number is controlled by the BouncePreScale register.

**Table 27-19. Bounce Enable Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [gn:0] | BounceEnable | Read/Write | Require INTR option |

Note: BounceEnable[I] is dedicated for pin[I]. I is ranging from 0 to gn.

### 27.4.2.18 BouncePreScale

The BouncePreScale register is an automatic register to indicate for the bounce timer. It can extend the PCLK to the BouncePreScale cycles. This register can be used to adjust the interrupt sample clock period in different machines. The reset value is 0x7D0, which means that if the APB clock frequency is 66 MHz, then the disbounce clock will be divided by (0x7D0+1) to 32.98 kHz. The programmer can adjust this register to fit different systems.

**Table 27-20. Bounce Clock Pre-scale Register**

| Bit | Name | Type | Comment |
|---|---|---|---|
| [23:0] | BouncePreScale | Read/Write | Require INTR option |

# Chapter 28

# Interrupt Controller

This chapter contains the following sections:


- 28.1 General Description
- 28.2 Features
- 28.3 Block Diagram
- 28.4 Interrupt Routing Table
- 28.5 Programming Model

## 28.1 General Description

The FTINTC010 interrupt controller provides both Fast Interrupt Request (FIQ) and Standard Interrupt Request (IRQ) signals to the microprocessor. The interrupt controller can handle up to 64 configurable standard interrupt sources and up to 64 fast interrupt sources. The output signals to the microprocessor can be configured as active high or active low.

## 28.2 Features

- Configurable controller to support up to 64 FIQ inputs
- Configurable controller to support up to 64 IRQ inputs
- Provides both-edge and level-triggered interrupt sources with positive and negative directions
- Independently configurable "ON" or "OFF" de-bounce circuit for each interrupt source
- Configures interrupt request signal to host controller to active high or active low

## 28.3 Block Diagram

Figure 28-1 shows the functional block diagram of the Faraday interrupt controller.

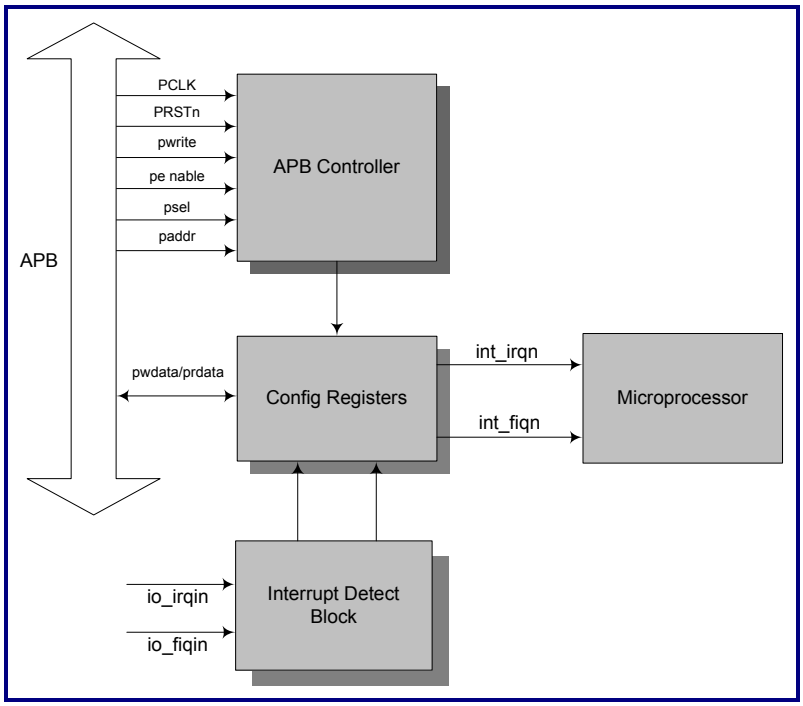



**Figure 28-1. Functional Block Diagram**

## 28.4 Interrupt Routing Table

The active high io_irqin and io_fiqin input signals indicate that a peripheral device is requesting an interrupt to the interrupt controller. The interrupt source assignment of io_irqin[63:0]/io_fiqin[63:0] is listed in Table 28-1.

**Table 28-1. Interrupt Routing Table (For IRQ and FIQ)**

| Bit | Device | Condition |
|---|---|---|
| [63:47] | - | Reserved |
| 46 | I2C_2 | Active high |
| 45 | I2C_1 | Active high |
| 44 | SATA100_1 | Active high, level-triggered |
| 43 | SATA100_0 | Active high, level-triggered |

| Bit | Device | Condition |
|---|---|---|
| 42 | DI3D_1 | Active high, level-triggered |
| 41 | DI3D_0 | Active high, level-triggered |
| 40 | VCAP_8 | Active high, level-triggered |
| 39 | VCAP_7 | Active high, level-triggered |
| 38 | VCAP_6 | Active high, level-triggered |
| 37 | VCAP_5 | Active high, level-triggered |
| 36 | VCAP_4 | Active high, level-triggered |
| 35 | VCAP_3 | Active high, level-triggered |
| 34 | VCAP_2 | Active high, level-triggered |
| 33 | VCAP_1 | Active high, level-triggered |
| 32 | VCAP_0 | Active high, level-triggered |
| 31 | JPEG | Active high, level-triggered |
| 30 | H.264 Decoder | Active high, level-triggered |
| 29 | H.264 Encoder | Active high, level-triggered |
| 28 | - | Reserved |
| 27 | DDRC_1 | Active high, level-triggered |
| 26 | DDRC_0 | Active high, level-triggered |
| 25 | IDE | Active high, level-triggered |
| 24 | LCDC | Active high, level-triggered |
| 23 | NANDC | Active high, level-triggered |
| 22 | UART_4 | Active high, level-triggered |
| 21 | UART_3 | Active high, level-triggered |
| 20 | UART_2 | Active high, level-triggered |
| 19 | AES | Active high, level-triggered |
| 18 | I2C_0 | Active high, level-triggered |
| 17 | PCI | Active high, level-triggered |
| 16 | WDT | Rising trigger |
| 15 | SDC | Active high, level-triggered |
| 14 | TIMER | Active high, level-triggered |
| 13 | GPIO | Active high, level-triggered |
| 12 | SCAR | Active high, level-triggered |
| 11 | SSP_1 | Active high, level-triggered |
| 10 | UART_1 | Active high, level-triggered |
| 9 | UART_0 | Active high, level-triggered |

| Bit | Device | Condition |
|---|---|---|
| 8 | PMU | Active high, level-triggered |
| 7 | SSP_2 | Active high, level-triggered |
| 6 | SSP_0 | Active high, level-triggered |
| 5 | USB HOST 2.0 | Polarity is programmable by using the USB HOST 2.0 controller. |
| 4 | USB OTG 2.0 | Polarity is programmable by using the USB OTG 2.0 controller. |
| 3 | GMAC | Active high, level-triggered |
| 2 | APBBRG | Active high, level-triggered |
| 1 | DMAC | Active high, level-triggered |
| 0 | AHBC_0 | Active high, level-triggered |

## 28.5 Programming Model

The following abbreviations are used for the register definitions:

R/W: Read/Write

WO: Write Only

RO: Read Only

### 28.5.1 Summary of Interrupt Controller Registers

The registers of the Faraday interrupt controller are listed in Table 28-2.

**Table 28-2. Summary of Interrupt Controller Registers**

| Address | R/W Type | Description | Default Value |
|---|---|---|---|
| +0x00 | RO | IRQ source register | 0x0 |
| +0x04 | R/W | IRQ enable register | 0x0 |
| +0x08 | WO | IRQ interrupt clear register | 0x0 |
| +0x0C | R/W | IRQ trigger-mode register | 0x0 |
| +0x10 | R/W | IRQ trigger-level register | 0x0 |
| +0x14 | RO | IRQ status register | 0x0 |
| +0x18 ~ +0x1C | - | Reserved | - |
| +0x20 | RO | FIQ source register | 0x0 |
| +0x24 | R/W | FIQ enable register | 0x0 |

| Address | R/W Type | Description | Default Value |
|---|---|---|---|
| +0x28 | WO | FIQ interrupt clear register | 0x0 |
| +0x2C | R/W | FIQ trigger-mode register | 0x0 |
| +0x30 | R/W | FIQ trigger-level register | 0x0 |
| +0x34 | RO | FIQ status register | 0x0 |
| +0x38 ~ +0x3C | - | Reserved | - |
| +0x50 | RO | Revision register | - |
| +0x54 | RO | Feature register of input number | - |
| +0x58 | RO | Feature register of IRQ de-bounce location | - |
| +0x5C | RO | Feature register of FIQ de-bounce location | - |
| +0x60 | RO | Extended IRQ source register | 0x0 |
| +0x64 | R/W | Extended IRQ enable register | 0x0 |
| +0x68 | WO | Extended IRQ interrupt clear register | 0x0 |
| +0x6C | R/W | Extended IRQ trigger-mode register | 0x0 |
| +0x70 | R/W | Extended IRQ trigger-level register | 0x0 |
| +0x74 | RO | Extended IRQ status register | 0x0 |
| +0x78 | RO | Extended feature register of IRQ de-bounce location | - |
| +0x7C | - | Reserved | - |
| +0x80 | RO | Extended FIQ source register | 0x0 |
| +0x84 | R/W | Extended FIQ enable register | 0x0 |
| +0x88 | WO | Extended FIQ interrupt clear register | 0x0 |
| +0x8C | R/W | Extended FIQ trigger-mode register | 0x0 |
| +0x90 | R/W | Extended FIQ trigger-level register | 0x0 |
| +0x94 | RO | Extended FIQ status register | 0x0 |
| +0x98 | RO | Extended feature register of FIQ de-bounce location | - |

### 28.5.2 Register Descriptions

Throughout the following sections on register descriptions, N indicates a configurable number from 1 to 64.

### 28.5.2.1 IRQ Source Register (+0x00)

For interrupt source 0 ~ 31

**Table 28-3. IRQ Source Register (+0x00)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | irqsrcreg | RO | Records the status of the input interrupt source<br>When set, IRQ input is asserted. |

### 28.5.2.2 IRQ Enable Register (+0x04)

For interrupt source 0 ~ 31

**Table 28-4. IRQ Enable Register (+0x04)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | enIRQint | R/W | Enables/disables the interrupt source of IRQ<br>0: Disables the interrupt source<br>1: Enables the interrupt source |

### 28.5.2.3 IRQ Interrupt Clear Register (+0x08)

For interrupt source 0 ~ 31

**Table 28-5. IRQ Interrupt Clear Register (+0x08)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | irqclear | WO | When set to '1,' this bit will automatically be reset to zero after clearing the interrupt status.<br>Note: This register takes effect only when the input interrupt sources are in the edge-triggered mode. |

### 28.5.2.4 IRQ Trigger Mode Register (+0x0C)

For interrupt source 0 ~ 31

**Table 28-6. IRQ Trigger Mode Register (+0x0C)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | irqtrigmode | R/W | Selects the type of interrupt source of IRQ<br>0: Level-triggered mode<br>1: Edge-triggered mode |

### 28.5.2.5 IRQ Trigger Level Register (+0x10)

For interrupt source 0 ~ 31

**Table 28-7. IRQ Trigger Level Register (+0x10)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | irqtriglevel | R/W | Selects the trigger level of interrupt source of IRQ<br>0: Active-high level trigger or rising-edge trigger<br>1: Active-low level trigger or falling-edge trigger |

### 28.5.2.6 IRQ Status Register (+0x14)

For interrupt source 0 ~ 31

**Table 28-8. IRQ Status Register (+0x14)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | irqstatus | RO | Records the IRQ status after masking.<br>0: No interrupt<br>1: Interrupt is asserted. |

### 28.5.2.7 FIQ Source Register (+0x20)

For interrupt source 0 ~ 31

**Table 28-9. FIQ Source Register (+0x20)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | fiqsrcreg | RO | Records the status of the input interrupt source<br>When set, FIQ input is asserted. |

### 28.5.2.8 FIQ Enable Register (+0x24)

For interrupt source 0 ~ 31

**Table 28-10. FIQ Enable Register (+0x24)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | enFIQint | R/W | Enables the interrupt source of FIQ.<br>0: Disables the interrupt source<br>1: Enables the interrupt source |

### 28.5.2.9 FIQ Interrupt Clear Register (+0x28)

For interrupt source 0 ~ 31

**Table 28-11. FIQ Interrupt Clear Register (+0x28)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | fiqclear | WO | When set to 1, this bit will be automatically reset to zero after clearing the interrupt status.<br>Note: This register takes effect only when the input interrupt sources are in the edge-trigger mode. |

### 28.5.2.10 FIQ Trigger Mode Register (+0x2C)

For interrupt source 0 ~ 31

**Table 28-12. FIQ Trigger Mode Register (+0x2C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | fiqtrigmode | R/W | Selects the type of interrupt source for FIQ<br>0: Level-triggered mode<br>1: Edge-triggered mode |

### 28.5.2.11 FIQ Trigger Level Register (+0x30)

For interrupt source 0 ~ 31

**Table 28-13. FIQ Trigger Level Register (+0x30)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | fiqtriglevel | R/W | Selects the trigger level of interrupt source for FIQ<br>0: Active-high level trigger or rising-edge trigger<br>1: Active-low level trigger or falling-edge trigger |

### 28.5.2.12 FIQ Status Register (+0x34)

For interrupt source 0 ~ 31

**Table 28-14. FIQ Status Register (+0x34)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:N] | - | - | Reserved |
| [N-1:0] | fiqstatus | RO | Records the FIQ status after masking<br>0: No interrupt<br>1: Interrupt is asserted. |

### 28.5.2.13 Revision Register (+0x50)

**Table 28-15. Revision Register (+0x50)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:24] | - | - | Reserved |
| [23:0] | intcrevision | RO | Records FTINTC010 IP revision |

### 28.5.2.14 Feature Register for Input Number (+0x54)

**Table 28-16. Feature Register for Input Number (+0x54)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:16] | - | - | Reserved |
| [15:8] | irq_number | RO | Record the supported IRQ pin number of FTINTC010 |
| [7:0] | fiq_number | RO | Record the supported FIQ pin number of FTINTC010 |

### 28.5.2.15 Feature Register for IRQ De-bounce Location (+0x58)

For interrupt source 0 ~ 31

**Table 28-17. Feature Register for IRQ De-bounce Location (+0x58)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | irq debounce | RO | Records IRQ interrupt source with de-bounce indication |

### 28.5.2.16 Feature Register for FIQ De-bounce Location (+0x5C)

For interrupt sources 0 ~ 31

**Table 28-18. Feature Register for FIQ De-bounce Location (+0x5C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | fiq debounce | RO | Records FIQ interrupt source with de-bounce indication |

### 28.5.2.17 Extended IRQ Source Register (+0x60)

For interrupt source 32 ~ 63

**Table 28-19. Extended IRQ Source Register (+0x60)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXIrqsrcreg | RO | Records the status of the input interrupt source<br>When set, the IRQ input is asserted. |

### 28.5.2.18 Extended IRQ Enable Register (+0x64)

For interrupt source 32 ~ 63

**Table 28-20. Extended IRQ Enable Register (+0x64)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXenIRQint | R/W | Enables/disables the interrupt source of IRQ<br>0: Disables the interrupt source<br>1: Enables the interrupt source |

### 28.5.2.19 Extended IRQ Interrupt Clear Register (+0x68)

For interrupt source 32 ~ 63

**Table 28-21. Extended IRQ Clear Register (+0x68)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXirqclear | WO | When set to 1, this bit will be automatically reset to zero after clearing the interrupt status.<br>Note: This register takes effect only when the input interrupt sources are in the edge-triggered mode. |

### 28.5.2.20 Extended IRQ Trigger Mode Register (+0x6C)

For interrupt source 32 ~ 63

**Table 28-22. Extended IRQ Trigger Mode Register (+0x6C)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXirqtrigmode | R/W | Select the type of interrupt source of IRQ<br>0: Level-triggered mode<br>1: Edge-triggered mode |

### 28.5.2.21 Extended IRQ Trigger Level Register (+0x70)

For interrupt source 32 ~ 63

**Table 28-23. Extended IRQ Trigger Level Register (+0x70)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXirqtriglevel | R/W | Select the trigger level of interrupt source of IRQ<br>0: Active-high level trigger or rising-edge trigger<br>1: Active-low level trigger or falling-edge trigger |

### 28.5.2.22 Extended IRQ Status Register (+0x74)

For interrupt source 32 ~ 63

**Table 28-24. Extended IRQ Status Register (+0x74)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXirqstatus | RO | Records the IRQ status after masking<br>0: No interrupt<br>1: Interrupt is asserted. |

### 28.5.2.23 Extended Feature Register for IRQ De-bounce Location (+0x78)

For interrupt source 32 ~ 63

**Table 28-25. Extended Feature Register for IRQ De-bounce Location (+0x78)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | EXirq debounce | RO | Records IRQ interrupt source with de-bounce indication |

### 28.5.2.24 Extended FIQ Source Register (+0x80)

For interrupt source 32 ~ 63

**Table 28-26. Extended FIQ Source Register (+0x80)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-33] | - | - | Reserved |
| [N-33:0] | EXfiqsrcreg | RO | Records the status of the input interrupt source<br>When set, the FIQ input is asserted. |

### 28.5.2.25 Extended FIQ Enable Register (+0x84)

For interrupt source 32 ~ 63

**Table 28-27. Extended FIQ Enable Register (+0x84)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXenFIQint | R/W | Enables/disables the interrupt source of FIQ<br>0: Disables the interrupt source<br>1: Enables the interrupt source |

### 28.5.2.26 Extended FIQ Interrupt Clear Register (+0x88)

For interrupt source 32 ~ 63

**Table 28-28. Extend FIQ Interrupt Clear Register (+0x88)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXfiqclear | WO | When set to 1, this bit will be automatically reset to zero after clearing the interrupt status.<br>Note: This register takes effect only when the input interrupt sources are in the edge-triggered mode. |

### 28.5.2.27 Extended FIQ Trigger Mode Register (+0x8C)

For interrupt source 32 ~ 63

**Table 28-29. Extended FIQ Trigger Mode Register (+0x8C)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | fiqtrigmode | R/W | Select the type of interrupt source for FIQ<br>0: Level-triggered mode<br>1: Edge-triggered mode |

### 28.5.2.28 Extended FIQ Trigger Level Register (+0x90)

For interrupt source 32 ~ 63

**Table 28-30. Extended FIQ Trigger Level Register (+0x90)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXfiqtriglevel | R/W | The trigger level of interrupt source of FIQ<br>0: Active-high level trigger or rising-edge trigger<br>1: Active-low level trigger or falling-edge trigger |

### 28.5.2.29 Extended FIQ Status Register (+0x94)

For interrupt source 32 ~ 63

**Table 28-31. Extended FIQ Status Register (+0x94)**

| Bit | Register Name | Type | Description |
|---|---|---|---|
| [31:N-32] | - | - | Reserved |
| [N-33:0] | EXfiqstatus | RO | Records the FIQ status after masking<br>0: No interrupt<br>1: Interrupt is asserted. |

### 28.5.2.30 Extended Feature Register for FIQ De-bounce Location (+0x78)

For interrupt source 32 ~ 63

**Table 28-32. Extended Feature Register for FIQ De-bounce Location (+0x78)**

| Bit | Name | Type | Description |
|---|---|---|---|
| [31:0] | EXfiq debounce | RO | Records FIQ interrupt source with de-bounce indication |

# Chapter 29

# Timer

This chapter contains the following sections:


- 29.1 General Description
- 29.2 Features
- 29.3 Block Diagram
- 29.4 Programming Model

## 29.1 General Description

The timer provides three independent sets of sub-timers. Each sub-timer can use either the internal system clock (PCLK) or the external clock (EXTCLK) to increase or decrease the counting. Two match registers are provided for each sub-timer. Whenever the value of the match registers equals any one value of the sub-timers, the timer interrupt is triggered immediately. The issuing of the timer interrupt can be decided by the register setting when an overflow occurs.

## 29.2 Features

- Three independent 32-bit timer programming models
- Internal or external clock source selection
- Interrupt can be issued upon overflow and time-up
- Each timer has two match registers
- Supports the incremental and decremental models

## 29.3 Block Diagram

Figure 29-1 shows the block diagram of the Faraday timer.

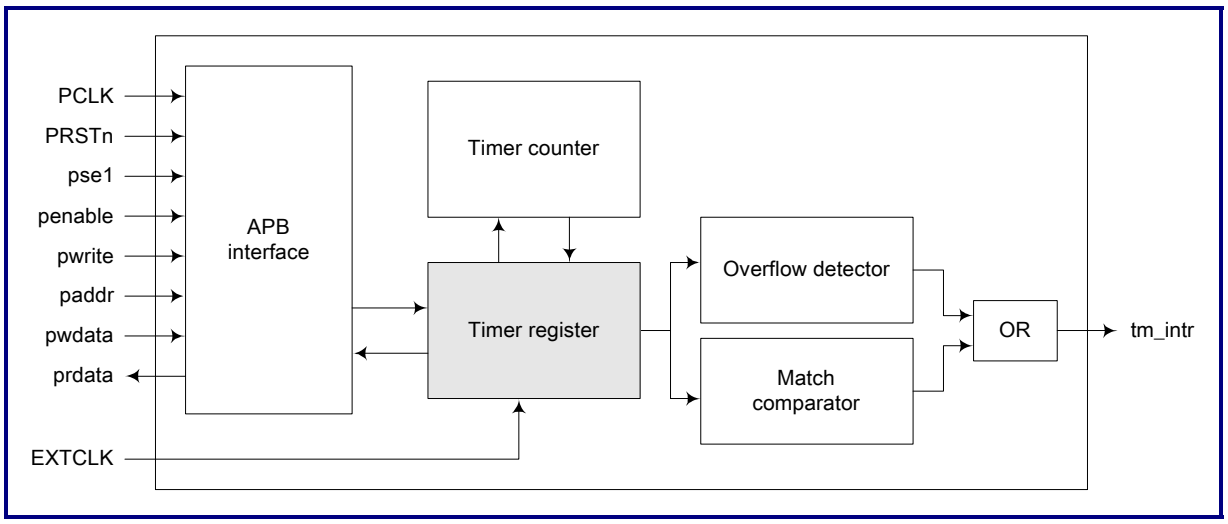



**Figure 29-1. Block Diagram**

## 29.4 Programming Model

### 29.4.1 Summary of Timer Registers

Table 29-1 shows the offset, type, width, reset value, name, and configuration option of each timer programming registers.

**Table 29-1. Summary of Timer Registers**

| Offset | Type | Width | Reset | Name | Configuration | Description |
|---|---|---|---|---|---|---|
| 0x00 | R/W | 32 | 0x-- | Tm1Counter | None | Timer1 counter |
| 0x04 | R/W | 32 | 0x-- | Tm1Load | None | Timer1 auto reload value |
| 0x08 | R/W | 32 | 0x-- | Tm1Match1 | None | Timer1 match value |
| 0x0C | R/W | 32 | 0x-- | Tm1Match2 | None | Timer1 match value |
| 0x10 | R/W | 32 | 0x-- | Tm2Counter | TM2 | Timer2 counter |
| 0x14 | R/W | 32 | 0x-- | Tm2Load | TM2 | Timer2 auto reload value |
| 0x18 | R/W | 32 | 0x-- | Tm2Match1 | TM2 | Timer2 match value |
| 0x1C | R/W | 32 | 0x-- | Tm2Match2 | TM2 | Timer2 match value |
| 0x20 | R/W | 32 | 0x-- | Tm3Counter | TM3 | Timer3 counter |
| 0x24 | R/W | 32 | 0x-- | Tm3Load | TM3 | Timer3 auto reload value |
| 0x28 | R/W | 32 | 0x-- | Tm3Match1 | TM3 | Timer3 match value |
| 0x2C | R/W | 32 | 0x-- | Tm3Match2 | TM3 | Timer3 match value |
| 0x30 | R/W | 12 | 0x0 | TmCR | None | Timer1, Timer2, Timger3 control register |
| 0x34 | R/W | 9 | 0x0 | IntrState | None | Interrupt State of FTTMR010 |
| 0x38 | R/W | 9 | 0x0 | IntrMask | None | Interrupt Mask of FTTMR010 |
| 0x3C | R | 32 | 0x-- | TmRevision | None | FTTMR010 revision number |

### 29.4.2 Register Description

The following sections describe the timer registers in detail.

#### 29.4.2.1 Tm1Counter, Tm2Counter, Tm3Counter

Tm1Counter, Tm2Counter, and Tm3Counter are the counter registers of Timer1, Timer2, and Timer3, respectively. When the timer is disabled, Tm(1 ~ 3)Counter will hold the value.

**Table 29-2. Tm1Counter, Tm2Counter, Tm3Counter Registers**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | Tm1Counter<br>Tm2Counter<br>Tm3Counter | Read/Write | Tm2Counter is enabled if the TM2 option is configured.<br>Tm3Counter is enabled if the TM3 option is configured. |

#### 29.4.2.2 Tm1Load, Tm2Load, Tm3Load

Tm1Load, Tm2Load, and Tm3Load are the auto-reload registers for Timer1, Timer2, and Timer3, respectively. Whenever an Timer(1~ 3) overflow occurs, the value of Tm(1 ~ 3)Load will be loaded into the counter (1 ~ 3) register. This value can be used to set the period between two overflows or underflows.

Note: An overflow indicates that the counter has exceeded the limit that can be supported. That is, the counter counts up to 0XFFFF_FFFF or down to 0X0000_0000.

**Table 29-3. Tm1Load, Tm2Load, Tm3Load Registers**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | Tm1Load<br>Tm2Load<br>Tm3Load | Read/Write | Tm2Counter is enabled if the TM2 option is configured.<br>Tm3Counter is enabled if the TM3 option is configured. |

### 29.4.2.3 Tm1Match1, Tm2Match1, Tm3Match1

Tm1Match1, Tm2Match1, and Tm3Match1 are the match registers of Timer1, Timer2, and Timer3, respectively. When counter(1 ~ 3) equals to Tm(1 ~ 3)Match1 and the Tm(1 ~ 3)Enable bit is set, tm(1 ~ 3)_intr will be triggered.

**Table 29-4. Tm1Match1, Tm2Match1, Tm3Match1 Registers**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | Tm1Match1<br>Tm2Match1<br>Tm3Match1 | Read/Write | Tm2Counter is enabled if the TM2 option is configured.<br>Tm3Counter is enabled if the TM3 option is configured. |

### 29.4.2.4 Tm1Match2, Tm2Match2, Tm3Match2

Tm1Match2, Tm2Match2, and Tm3Match2 are the match registers of Timer1, Timer2, and Timer3, respectively. When counter(1 ~ 3) equals to the Tm(1 ~ 3)Match2 and Tm(1 ~ 3)Enable bits, then tm(1~3)_intr will be triggered. The counter will keep counting, and reload will not occur.

**Table 29-5. Tm1Match2, Tm2Match2, Tm3Match2 Registers**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | Tm1Match2<br>Tm2Match2<br>Tm3Match2 | Read/Write | Tm2Counter is enabled if the TM2 option is configured.<br>Tm3Counter is enabled if the TM3 option is configured. |

### 29.4.2.5 TmCR

TmCR is the control register of the Faraday timer that controls the timer enable/disable function, clock source selection, and overflow mode. When Tm1Enable is set to 0, tm1_intr will never be triggered and the value of counter1 will be held. When Tm1Clock in TmCR is set to 0, counter1 will use PCLK as the clock source; otherwise, counter1 will use EXT1CLK as the clock source. When Tm1OFEnable is set, tm1_intr will be triggered once the timer1 overflow occurs. Otherwise, tm1_intr is only triggered when the value of counter1 equals the value of Tm1Match1 or Tm1Match2.

Table 29-6 is a summary of the bits and functions of TmCR.

**Table 29-6. Timer Control Register**

| Bit | Name | Description |
|---|---|---|
| 0 | Tm1Enable | Timer1 enable bit<br>0: Disable<br>1: Enable |
| 1 | Tm1Clock | Timer1 clock source<br>0: PCLK<br>1: EXT1CLK |
| 2 | Tm1OFEnable | Timer1 overflow interrupt enable bit<br>0: Disable<br>1: Enable |
| 3 | Tm2Enable | Timer2 enable bit<br>0: Disable<br>1: Enable |
| 4 | Tm2Clock | Timer2 clock source<br>0: PCLK<br>1: EXT2CLK |
| 5 | Tm2OFEnable | Timer2 overflow interrupt enable bit<br>0: Disable<br>1: Enable |
| 6 | Tm3Enable | Timer3 enable bit<br>0: Disable<br>1: Enable |
| 7 | Tm3Clock | Timer3 clock source<br>0: PCLK<br>1: EXT3CLK |
| 8 | Tm3OFEnable | Timer3 overflow interrupt enable bit<br>0: Disable<br>1: Enable |

| Bit | Name | Description |
|---|---|---|
| 9 | Tm1UpDown | Timer1 up or down count<br>0: Down count<br>The Tm1Counter decreases to 0 from its initial value. When it underflows, Tm1Counter will auto-reload the value of Tm1Load into Tm1Counter.<br>1: Up count<br>The Tm1Counter increases to 0xffff from its initial value. When it overflows, Tm1Counter will auto-reload the value of Tm1Load into Tm1Counter. |
| 10 | Tm2UpDown | Timer2 up or down count<br>0: Down count<br>The Tm2Counter decreases to 0 from its initial value. When it underflows, Tm2Counter will auto-reload the value of Tm2Load into Tm2Counter.<br>1: Up count<br>The Tm2Counter increases to 0xffff from its initial value. When it overflows, Tm2Counter will auto-reload the value of Tm2Load into Tm2Counter. |
| 11 | Tm3UpDown | Timer3 up or down count<br>0: Down count<br>The Tm3Counter decreases to 0 from its initial value. When it underflows, Tm1Counter will auto-reload the value of Tm3Load into Tm3Counter.<br>1: Up count<br>The Tm3Counter increases to 0xffff from its initial value. When it overflows, Tm3Counter will auto-reload the value of Tm3Load into Tm3Counter. |

### 29.4.2.6 IntrState

The IntrState register is the interrupt state register of FTTMR010. When tmr_intr is asserted, the CPU must check which interrupt has occurred by reading the IntrState register. Each bit of the IntrState register must indicate a corresponding interrupt that has occurred. All the bits of IntrState must be cleared by firmware. If some bits of pwdata are set, the related bits of IntrState will be cleared when the user writes to the IntrState address.

**Table 29-7. Timer Interrupt State Register**

| Bit | Name | Description |
|---|---|---|
| 0 | Tm1Match1 | Tm1Match1 interrupt<br>0: No effect<br>1: Tmr1 counter value equals to the value in the Tm1Match1 register. |
| 1 | Tm1Match2 | Tm1Match2 interrupt<br>0: No effect<br>1: Tmr1 counter value equals to the value in the Tm1Match2 register. |

| Bit | Name | Description |
|---|---|---|
| 2 | Tm1Overflow | Tm1Overflow interrupt<br>0: No effect<br>1: Tmr1 counter overflow |
| 3 | Tm2Match1 | Tm2Match1 interrupt<br>0: No effect<br>1: Tmr2 counter value equals to the value in the Tm2Match1 register. |
| 4 | Tm2Match2 | Tm2Match2 interrupt<br>0: No effect<br>1: Tmr2 counter value equals to the value in the Tm2Match2 register. |
| 5 | Tm2Overflow | Tm2Overflow interrupt<br>0: No effect<br>1: Tmr2 counter overflow |
| 6 | Tm3Match1 | Tm3Match1 interrupt<br>0: No effect<br>1: Tmr3 counter value equals to the value in the Tm3Match1 register. |
| 7 | Tm3Match2 | Tm3Match2 interrupt<br>0: No effect<br>1: Tmr3 counter value equals to the value in theTm3Match2 register. |
| 8 | Tm3Overflow | Tm3Overflow interrupt<br>0: No effect<br>1: Tmr3 counter overflow |

### 29.4.2.7 IntrMask

The IntrMask register is an interrupt mask register. If one bit of the IntrMask register is set, the corresponding bit of IntrState will be masked and will never assert tmr_intr even though the corresponding bit of IntrState is logic one.

**Table 29-8. Timer Interrupt Mask Register**

| Bit | Name | Description |
|---|---|---|
| 0 | MTm1Match1 | Mask Tm1Match1 interrupt<br>0: No effect<br>1: Tm1Match1 in IntrState will be masked. |
| 1 | MTm1Match2 | Mask Tm1Match2 interrupt<br>0: No effect<br>1: Tmr1Match2 in IntrState will be masked. |
| 2 | MTm1Overflow | Mask Tm1Overflow interrupt<br>0: No effect<br>1: Tmr1Overflow in IntrState will be masked. |
| 3 | MTm2Match1 | Mask Tm2Match1 interrupt<br>0: No effect<br>1: Tm2Match1 in IntrState will be masked. |
| 4 | MTm2Match2 | Mask Tm2Match2 interrupt<br>0: No effect<br>1: Tmr2Match2 in IntrState will be masked. |
| 5 | MTm2Overflow | Mask Tm2Overflow interrupt<br>0: No effect<br>1: Tmr2Overflow in IntrState will be masked. |
| 6 | MTm3Match1 | Mask Tm3Match1 interrupt<br>0: No effect<br>1: Tm3Match1 in IntrState will be masked. |
| 7 | MTm3Match2 | Mask Tm3Match2 interrupt<br>0: No effect<br>1: Tmr3Match2 in IntrState will be masked. |
| 8 | MTm3Overflow | Mask Tm3Overflow interrupt<br>0: No effect<br>1: Tmr3Overflow in IntrState will be masked. |

### 29.4.2.8 TmRevision

The TmRevision register contains the revision information of FTTMR010.

**Table 29-9. TmRevision Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | TmRevision | Read | The revision number of FTTMR010 |

# Chapter 30

# Timer With PWM Function

This chapter contains the following sections:


- 30.1 General Description
- 30.2 Features
- 30.3 Block Diagram
- 30.4 Programming Model
- 30.5 Basic Timer Operation

## 30.1 General Description

FTPWMTMR010 is an APB device which provides up to eight independent sets of timers. Each timer can use either the internal system clock (The PCLK of APB) or external clock. These timers can be used to generate internal interrupts to the CPU. They can also be used to trigger DMA transfers. In addition, each timer supports PWM (Pulse Width Modulation) function, which can generate PWM signals for motor control or power level control.

## 30.2 Features

- Supports the AMBA 2.0 APB bus
- Supports up-to 8 independent 32-bit timers
- Internal or external clock source selection for each timer
- Supports PWM generation for each timer
- Programmable duty control and polarity of PWM
- Supports the Auto-reload mode and the One-shot pulse mode
- Dead-zone generator on PWM outputs
- Supports timer-based DMA transfer request

## 30.3 Block Diagram

Figure 30-1 shows the pin diagram of FTPWMTMR010.

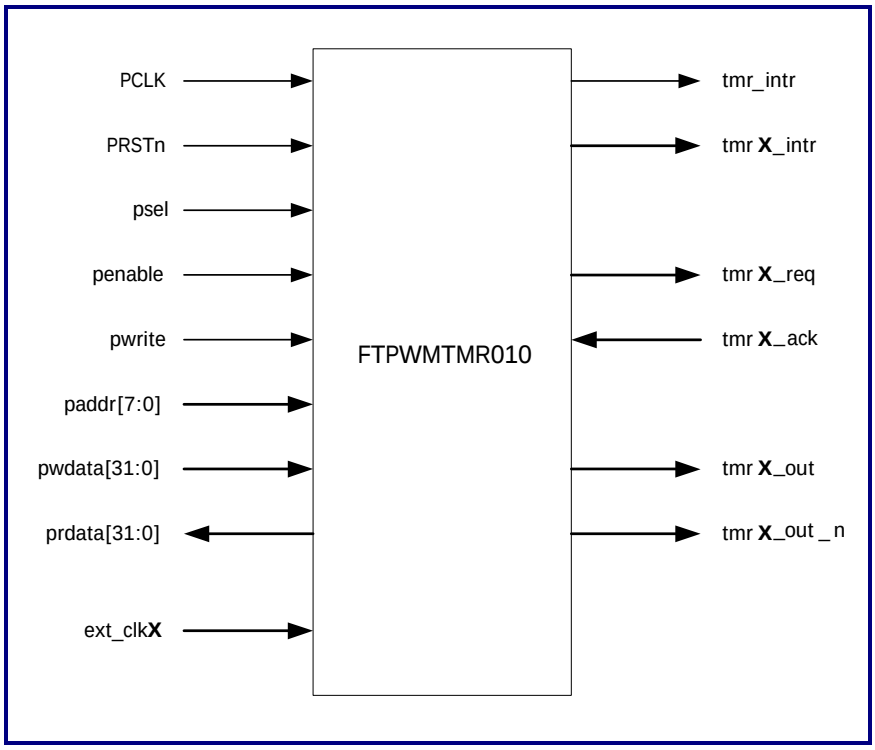



**Figure 30-1. Pin Diagram**

## 30.4 Programming Model

### 30.4.1 Summary of Timer Registers

Table 30-1 lists the offsets, types, widths, reset values, names, and descriptions of timer registers. Besides global registers such as INT_STAT, INT_CTRL, and TMR_REV are shared by all timers, each timer has its own register set. The settings of these registers depend on the presence of their corresponding timers. Users should not access the registers of non-exist timers.

**Table 30-1. Summary of Timer Registers**

| Offset | Name | Type | Width | Reset | Description |
|---|---|---|---|---|---|
| 0x00 | INT_CSTAT | R/WC | 8 | 0x0 | Interrupt status and control register of FTTMR020 |
| 0x10 | T1_CTRL | R/W | 16 | 0x0 | Timer 1 control register |
| 0x14 | T1_CNTB | R/W | 32 | 0x0 | Timer 1 count buffer register |
| 0x18 | T1_CMPB | R/W | 32 | 0x0 | Timer 1 compare buffer register |
| 0x1C | T1_CNTO | R | 32 | 0x0 | Timer 1 observation register |
| 0x20 | T2_CTRL | R/W | 16 | 0x0 | Timer 2 control register |
| 0x24 | T2_CNTB | R/W | 32 | 0x0 | Timer 2 count buffer register |
| 0x28 | T2_CMPB | R/W | 32 | 0x0 | Timer 2 compare buffer register |
| 0x2C | T2_CNTO | R | 32 | 0x0 | Timer 2 observation register |
| 0x30 | T3_CTRL | R/W | 16 | 0x0 | Timer 3 control register |
| 0x34 | T3_CNTB | R/W | 32 | 0x0 | Timer 3 count buffer register |
| 0x38 | T3_CMPB | R/W | 32 | 0x0 | Timer 3 compare buffer register |
| 0x3C | T3_CNTO | R | 32 | 0x0 | Timer 3 observation register |
| 0x40 | T4_CTRL | R/W | 16 | 0x0 | Timer 4 control register |
| 0x44 | T4_CNTB | R/W | 32 | 0x0 | Timer 4 count buffer register |
| 0x48 | T4_CMPB | R/W | 32 | 0x0 | Timer 4 compare buffer register |
| 0x4C | T4_CNTO | R | 32 | 0x0 | Timer 4 observation register |
| 0x50 | T5_CTRL | R/W | 16 | 0x0 | Timer 5 control register |
| 0x54 | T5_CNTB | R/W | 32 | 0x0 | Timer 5 count buffer register |
| 0x58 | T5_CMPB | R/W | 32 | 0x0 | Timer 5 compare buffer register |
| 0x5C | T5_CNTO | R | 32 | 0x0 | Timer 5 observation register |
| 0x60 | T6_CTRL | R/W | 16 | 0x0 | Timer 6 control register |
| 0x64 | T6_CNTB | R/W | 32 | 0x0 | Timer 6 count buffer register |

| Offset | Name | Type | Width | Reset | Description |
|---|---|---|---|---|---|
| 0x68 | T6_CMPB | R/W | 32 | 0x0 | Timer 6 compare buffer register |
| 0x6C | T6_CNTO | R | 32 | 0x0 | Timer 6 observation register |
| 0x70 | T7_CTRL | R/W | 16 | 0x0 | Timer 7 control register |
| 0x74 | T7_CNTB | R/W | 32 | 0x0 | Timer 7 count buffer register |
| 0x78 | T7_CMPB | R/W | 32 | 0x0 | Timer 7 compare buffer register |
| 0x7C | T7_CNTO | R | 32 | 0x0 | Timer 7 observation register |
| 0x80 | T8_CTRL | R/W | 16 | 0x0 | Timer 8 control register |
| 0x84 | T8_CNTB | R/W | 32 | 0x0 | Timer 8 count buffer register |
| 0x88 | T8_CMPB | R/W | 32 | 0x0 | Timer 8 compare buffer register |
| 0x8C | T8_CNTO | R | 32 | 0x0 | Timer 8 observation register |
| 0x90 | TMR_REV | R | 32 | 0x100 | FTTMR020 revision |

### 30.4.2 Register Description

The following sections describe the timer registers in details.

#### 30.4.2.1 Interrupt Status and Control Register: INT_CSTAT

The INT_CSTAT register shows individual interrupt status of each timer. If the global interrupt of FTPWMTMR010 is used by the system, the CPU can check the timer that issues interrupt by reading the INT_CSTAT register. The interrupt enable/disable control of each timer can also be set in this register.

| Bit | Name | Type | Reset | Description |
|---|---|---|---|---|
| 0 | TM1_INT_STAT | R/WC | 0 | Timer 1 interrupt status<br>Clear by writing '1' on this bit.<br>0: No effect<br>1: Counter value of Timer 1 reaches zero |
| 1 | TM2_INT_STAT | R/WC | 0 | Timer 2 interrupt status<br>Clear by writing '1' on this bit.<br>0: No effect<br>1: Counter value of Timer 2 reaches zero |

| Bit | Name | Type | Reset | Description |
|---|---|---|---|---|
| 2 | TM3_INT_STAT | R/WC | 0 | Timer 3 interrupt status<br>Clear by writing ‘1’ on this bit.<br>0: No effect<br>1: Counter value of Timer 3 reaches zero |
| 3 | TM4_INT_STAT | R/WC | 0 | Timer 4 interrupt status<br>Clear by writing ‘1’ on this bit.<br>0: No effect<br>1: Counter value of Timer 4 reaches zero |
| 4 | TM5_INT_STAT | R/WC | 0 | Timer 5 interrupt status<br>Clear by writing ‘1’ on this bit.<br>0: No effect<br>1: Counter value of Timer 5 reaches zero |
| 5 | TM6_INT_STAT | R/WC | 0 | Timer 6 interrupt status<br>Clear by writing ‘1’ on this bit.<br>0: No effect<br>1: Counter value of Timer 6 reaches zero |
| 6 | TM7_INT_STAT | R/WC | 0 | Timer 7 interrupt status<br>Clear by writing ‘1’ on this bit.<br>0: No effect<br>1: Counter value of Timer 7 reaches zero |
| 7 | TM8_INT_STAT | R/WC | 0 | Timer 8 interrupt status<br>Clear by writing ‘1’ on this bit.<br>0: No effect<br>1: Counter value of Timer 8 reaches zero |
| [31:8] | - | - | - | Reserved. |

### 30.4.3 Timer Control Register: TMX_CTRL (X = 1, 2, …, 8)

This register controls the behavior of a timer, including start, update, inverted output, auto-reload, DMA, and dead-zone.

| Bit | Name | Type | Reset | Description |
|---|---|---|---|---|
| 0 | TMX_CLK_SRC | R/W | 0 | TimerX clock source<br>0: PCLK<br>1: ext_clk |
| 1 | TMX_START | R/W | 0 | TimerX start/stop<br>0: Stop<br>1: Start |
| 2 | TMX_UPDATE | R/W | 0 | TimerX update TMX_CNTB, TMX_CMPB. Write '1' to this register will trigger a manual update. This bit will be auto-cleared after the operation.<br>0: No operation<br>1: Manual updates |
| 3 | TMX_OUT_INV | R/W | 0 | TimerX output inverter on/off<br>0: Inverter off<br>1: Inverter on |
| 4 | TMX_AUTO | R/W | 0 | TimerX auto-reload on/off<br>0: One-shot<br>1: Interval mode (Auto-reload) |
| 5 | TMX_INT_EN | R/W | 0 | TimerX Interrupt mode enable/disable<br>0: Disable<br>1: Enable |
| 6 | TMX_INT_MO | R/W | 0 | TimerX Interrupt mode<br>0: Level (Clear interrupt by writing '1' on TMX_INT_STAT)<br>1: Pulse |
| 7 | TMX_DMA_EN | R/W | 0 | TimerX DMA request mode enable/disable<br>0: Disable<br>1: Enable |
| 8 | TMX_PWM_EN | R/W | 0 | TimerX PWM function enable/disable<br>0: Disable<br>1: Enable |
| [23:9] | - | - | - | Reserved. |
| [31:24] | TMX_DZ | R/W | 0 | TimerX dead-zone length |

### 30.4.4 Timer Counting Value Buffer Register: TMX_CNTB (X=1, 2, …, 8)

This register stores the start value of the counter. The counter in the timer will load this value when:

1. Manual update triggered by TM**X**_CTRL[TM**X**_UPDATE], or
2. Auto-reload is enabled and counter value reaches zero

| Bit | Name | Type | Reset | Description |
|---|---|---|---|---|
| [31:0] | TMX_CNTB | R/W | 0 | Start value of TimerX counter |

### 30.4.5 Timer Counter Compare Buffer Register: TMX_CMPB (X = 1, 2, …, 8)

This register stores the compare value of the counter. When the counter value equals to the compare value, the level of timer output tmr**X**_out will be changed. The comparator in the timer will load this value when :

1. Manual update triggered by TM**X**_CTRL[TM**X**_UPDATE], or
2. Auto-reload is enabled and counter value reaches zero

| Bit | Name | Type | Reset | Description |
|---|---|---|---|---|
| [31:0] | TMX_CMPB | R/W | 0 | Compare value of TimerX |

### 30.4.6 Timer Count Observation Register: TMX_CNTO (X = 1, 2, …, 8)

The CPU can observe the counter value by reading this register.

| Bit | Name | Type | Reset | Description |
|---|---|---|---|---|
| [31:0] | TMX_CNTO | R | 0 | Compare value of TimerX |

### 30.4.7 Revision Register: TMR_REV

Users can get the revision of TIMER by reading this register.

| Bit | Name | Type | Reset | Description |
|---|---|---|---|---|
| [31:0] | REV | R | 0x100 | The revision of FTTMR020. |

## 30.5 Basic Timer Operation

The following procedures of timer operation are depicted in Figure 5-1.

1. Enable the Auto-reload feature (TM**X**_CTRL[TM**X**_AUTO]). Set the TMX_CNTB as 160 (50 + 110) and the TMX_CMPB as 110. Set the manual update bit (TM**X**_CTRL[TM**X**_UPDATE]) and inverter bit on/off (TM**X**_CTRL[TM**X**_OUT_INV]). The manual update bit sets the values of the counter and comparator to TM**X**_CNTB and TM**X**_CMPB.
2. Start the timer by setting the start bit and manual update bit off.
3. When the counter has the same value as the comparator, the logic level of tmr**X**_out changes from low to high.
4. When the counter reaches '0', the counter reloads automatically with TM**X**_CNTB. At the same time, the interrupt request is generated, and tmr**X**_out changes from high to low.
5. In the ISR (Interrupt Service Routine), TM**X**_CNTB and TM**X**_CMPB are set as 80 (20 + 60) and 60, which is used for next duration.
6. When the counter has the same value as the comparator, the logic level of tmr**X**_out changes from low to high.
7. When the counter reaches '0', the counter reloads automatically with TM**X**_CNTB. At the same time, the interrupt request is generated, and tmr**X**_out changes from high to low.
8. In the ISR, auto-reload and interrupt request are disabled to stop the timer.
9. When the counter has the same value as the comparator, the logic level of tmr**X**_out changes from low to high.
10. Even when the counter reaches '0', the counter no longer reloads and the timer stops since the Auto-reload feature is disabled.
11. No interrupt request is generated.

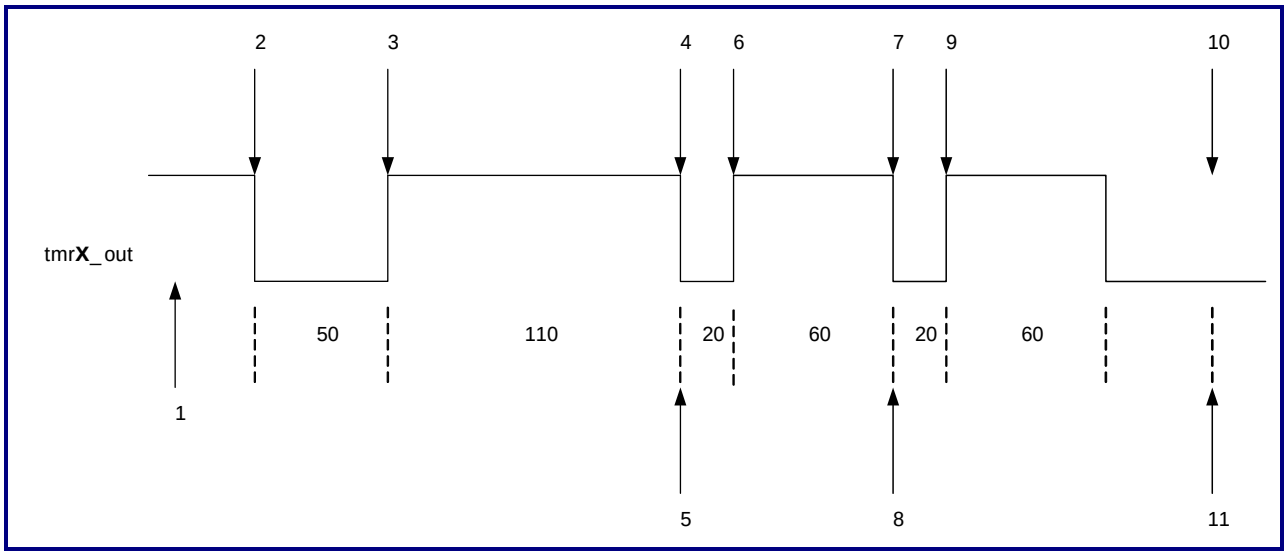



**Figure 30-2. Example of Timer Operation**

### 30.5.1 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. Consequently, a current timer operation can be completed successfully even if the new time value is set.

The timer value can be written into TM**X**_CNTB (Timer count buffer register) and the current counter value of the timer can be read from TM**X**_CNTO (Timer count observation register). When TM**X**_CNTB reads, the value read is the reload value for the next timer duration, not the current state of the counter.

The auto-reload feature is the operation that copies TM**X**_CNTB into the counter when the counter reaches '0' and when the Auto-reload feature is enabled. Under the condition that the counter reaches '0' and the Auto-reload feature is disabled, the counter will stop operating. This is called "one-shot operation".

Figure 30-3 shows an example of double buffering feature.

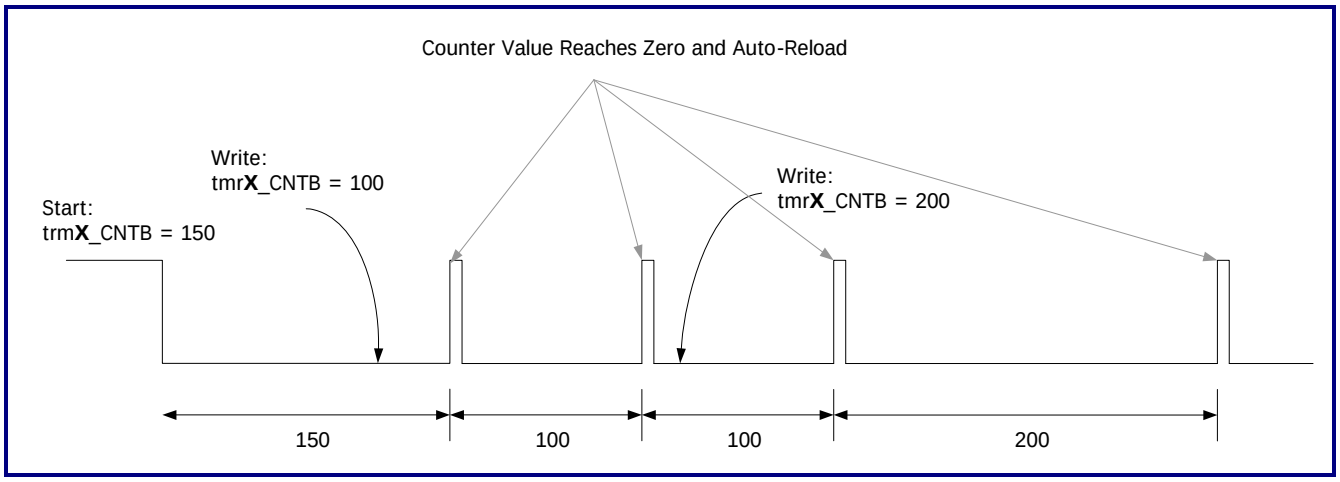



**Figure 30-3. Double Buffering Function**

## 30.5.2 PWM Function

The PWM function can be implemented by using TM**X**_CMPB. When the output inverter, TM**X**_CTRL[TM**X**_OUT_INV], is disabled, the timer output, tmr**X**_out, will go to low level whenever the manual update or auto-reload of the counter occurs; the output will go to high level whenever the counter value equals to the compare value. This mechanism implies that the PWM period is determined by TM**X**_CNTB and the pulse width is determined by TM**X**_CMPB.

For higher PWM values, increase the TM**X**_CMPB value. For lower PWM values, decrease the TM**X**_CMPB value. However, if the output inverter is enabled, the increment/decrement may be opposite.

Because of the double buffering feature, TM**X**_CMPB, for a next PWM cycle, can be written into any point of the current PWM cycle by ISR.

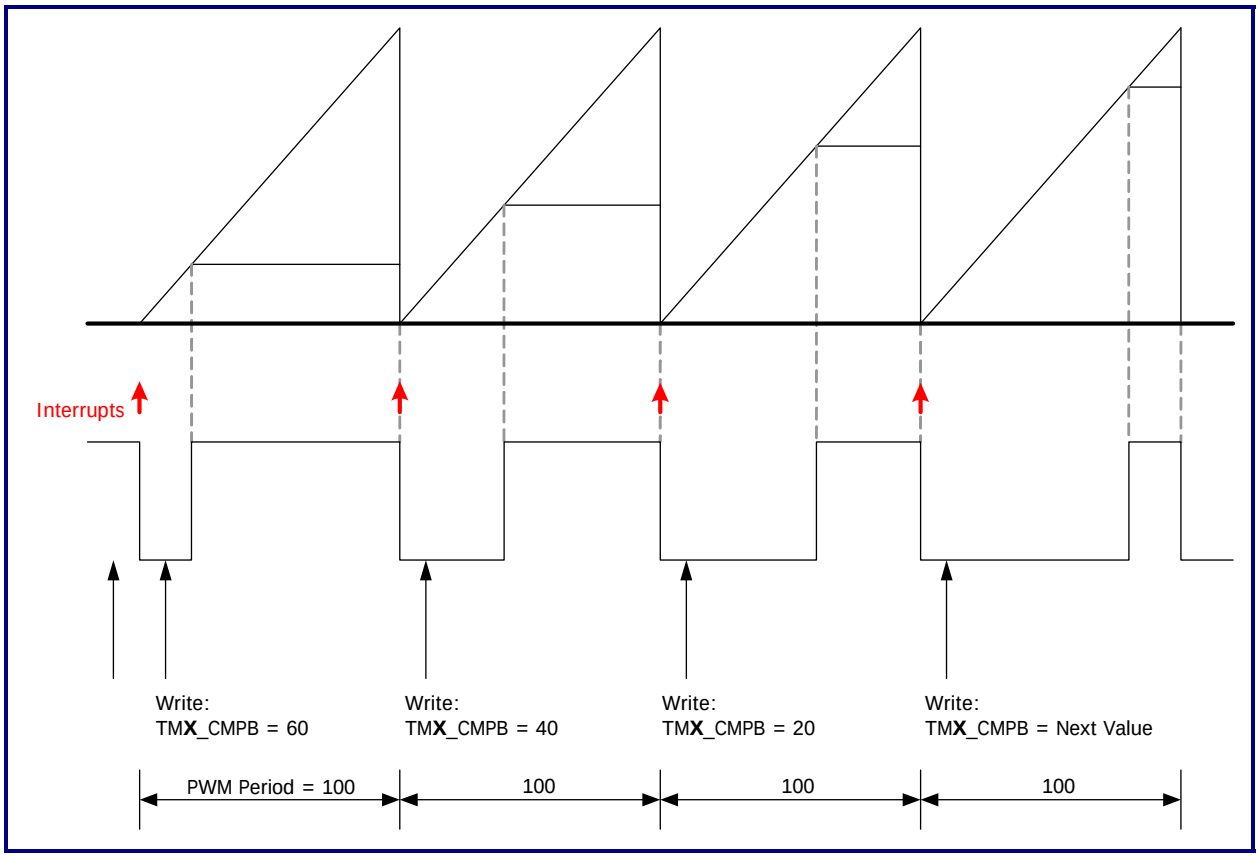



**Figure 30-4. Example of PWM**

### 30.5.3 Dead-zone Generation

The dead-zone 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 prohibits the two switching devices to simultaneously turn on even for a short period of time.

Figure 5-4 on the next page shows the waveform when the dead-zone feature is enabled. Where tmr**X**_out is the PWM output and tmr**X**_out_n is the inversion of tmr**X**_out. When the dead-zone feature is enabled, the output wave-form of tmr**X**_out and tmr**X**_out_n will be tmr**X**_out_DZ and tmr**X**_out_n_DZ. These two signals can never be turned on simultaneously by the dead-zone interval. For functional correctness, the dead-zone length must be set smaller than the compare counter value; otherwise, pulses that with a pulse width of less than the dead-zone will vanish.

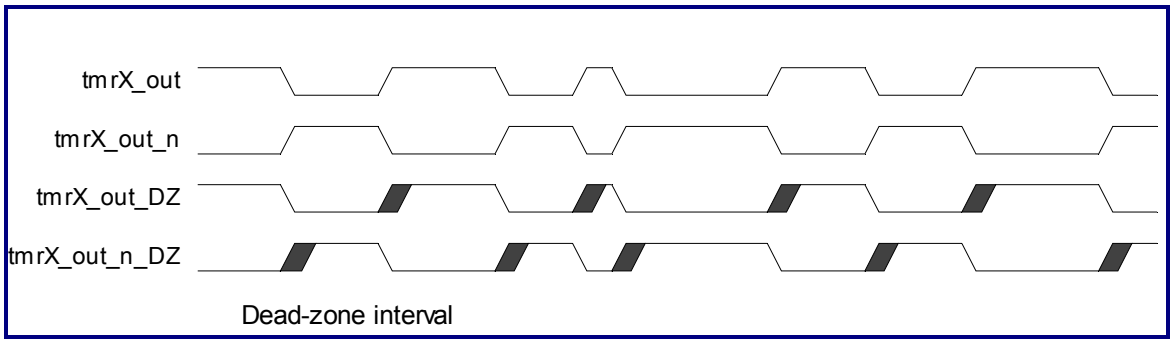



**Figure 30-5. Waveform of Enabled Dead-zone Feature**

### 30.5.4 DMA Request Mode

Instead of sending an interrupt at the end of the down-counter cycle, a timer can be configured to send a DMA request signal to one of the DMA channels. This mode allows the occurrence of DMA transfers between a source and destination at regular intervals. The idea is not to have the timer as the source or destination, but rather to have the timer simply control the timing of data movement between two other devices.

The DMA request signal, tmr**X**_req, will be kept active high by the timer until the timer receives an ACK signal from the DMA unit. When the timer receives an ACK signal, the request signal will become inactive.

If a timer is a DMA request mode that is enabled, it will not issue any interrupt, even when the interrupt mode is on. Figure 30-6 shows how the DMA request remain active before the ACKJ signal is sent by the DMA.

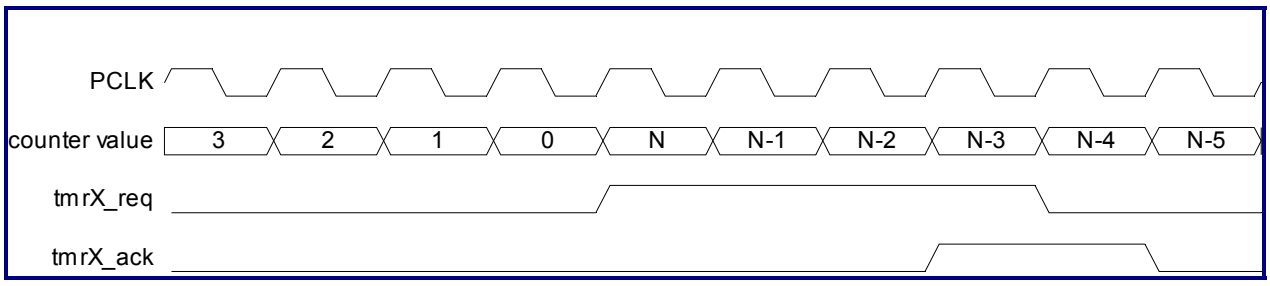



**Figure 30-6. DMA Request and ACK**

### 30.5.5 Initialization/Application Information

When PCLK = 66 MHz, the minimum resolution = 0.015 μs, and the maximum interval = 64.42s (TMX_CNTB = 232 - 1)

For purposes of synchronization, the clocks, PCLK and ext_clkX, must maintain the following relationship: ext_clkX cycle time/PCLK cycle time > 2

# Chapter 31

# Watchdog Timer

This chapter contains the following sections:


- 31.1 General Description
- 31.2 Features
- 31.3 Block Diagram
- 31.4 Programming Model

## 31.1 General Description

The WatchDog Timer (WDT) is used to prevent the system from infinite looping if the software is trapped in the deadlock. In the normal operation, the user restarts the WDT at the regular intervals before the counter counts down to 0. If the counter does reach 0, the WDT generates one or a combination of the signals, system reset, system interrupt, or external interrupt to reset the system, interrupt the system, or interrupt an external device correspondingly.

## 31.2 Features

- During the timeout, the outputs are one or a combination of the following signals:
  - System reset
  - System interrupt
  - External interrupt
- 32-bit down counter
- Internal or external clock source selection
- A variable time-out period of reset
- Access protection

## 31.3 Block Diagram

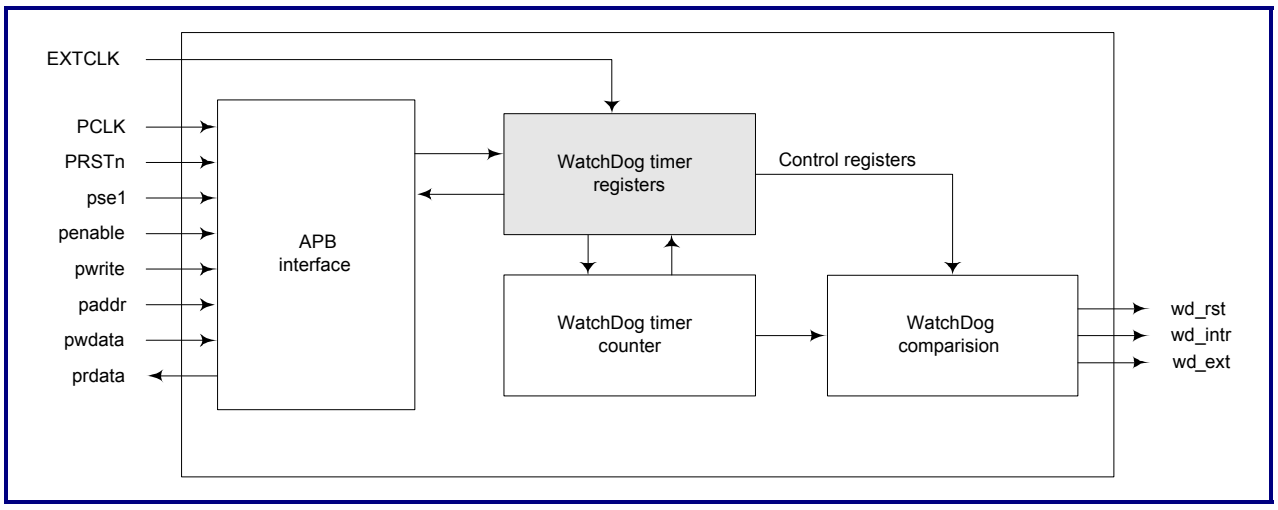



**Figure 31-1. Block Diagram of the WatchDog Timer**

## 31.4 Programming Model

### 31.4.1 Summary of the WatchDog Timer Register

Table 31-1 shows the definitions of each programming register of the WatchDog timer.

**Table 31-1. Summary of WatchDog Timer Register**

| Offset | Type | Width | Reset Value | Name | Config. | Description |
|---|---|---|---|---|---|---|
| 0x00 | R | 32 | 0x3EF1480 | WdCounter | None | The WatchDog timer counter register |
| 0x04 | R/W | 32 | 0x3EF1480 | WdLoad | None | The WatchDog timer counter auto reload register<br>The auto-reload register is set to 0x3EF1480 as the default. |
| 0x08 | W | 16 | 0x0000 | WdRestart | None | The WatchDog timer counter restart register<br>When writing 0x5AB9 to this register, the WatchDog timer will automatically reload the WdLoad to Wdcounter and restart the counting. |
| 0x0C | R/W | 5 | 0x0 | WdCR | None | The WatchDog timer control register |
| 0x10 | R | 1 | 0x0 | WdStatus | None | The WatchDog timer status register<br>This bit is set when the counter reaches 0.<br>0: Does not reach 0<br>1: Reach 0 |
| 0x14 | W | 1 | 0x0 | WdClear | None | The WatchDog timer is cleared.<br>Writing 1 or 0 to this register will clear the WdStatus. |
| 0x18 | R/W | 8 | 0xFF | WdIntrlen | None | The WatchDog timer interrupt length<br>This register controls the length of wd_rst, wd_intr, and wd_ext. The default value is 0xFF. |
| 0x1C | R | 32 | 0x-- | WdRevision | None | The revision number of FTWDT010 |

### 31.4.2 Register Description

The following sections describe the detailed information of the WatchDog timer registers.

### 31.4.2.1 WdCounter (Offset = 0x00)

**Table 31-2. WdCounter Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | WdCounter | R | The WdCounter register contains the current counter value. When reset, the WdCounter register is set to 0x3EF1480. After the programmer writes 0x5AB9 to WdRestart, the value of WdLoad will be loaded into WdCounter. The WdCounter register starts to decrease once WdCR[0], the WatchDog timer enable bit, is set. If the WatchDog timer is disabled, WdCounter will hold the value. If WdCR[4] is set, the register is driven by an external clock, and WdCounter will decrease in the EXTCLK frequency. This register is read only. |

### 31.4.2.2 WdLoad (Offset = 0x04)

**Table 31-3. WdLoad Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | WdLoad | R/W | The WdLoad register contains the value which will be loaded into WdCounter. When reset or restarted, the value of WdLoad will be automatically loaded into the WdCounter register. The reset value of WdLoad is 0x3EF1480. |

### 31.4.3 WdRestart (Offset = 0x08)

**Table 31-4. WdRestart Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | WdRestart | W | The WdRestart register is used to avoid the unexpected counting. If the programmer writes 0x5AB9 to this register, the WatchDog timer counter will load WdLoad into the WdCounter register and the WatchDog timer counter will restart to decrease. After finishing the write cycle, WdRestart will automatically be reset to 0. |

### 31.4.3.1 WdCR (Offset = 0x0C)

WdCR is the WatchDog timer control register. The following table lists the definitions of enable bits, reset bits, interrupt bits, external enable bits, and clock source bits.

**Table 31-5. WdCR Register**

| Bit | Name | Reset | Description |
|---|---|---|---|
| 0 | WdEnable | 0x0 | The WatchDog timer enable bit<br>0: Disable<br>1: Enable |
| 1 | WdRst | 0x0 | The WatchDog timer system reset enable bit<br>0: Disable<br>1: Enable |
| 2 | WdIntr | 0x0 | The WatchDog timer system interrupt enable bit<br>0: Disable<br>1: Enable |
| 3 | WdExt | 0x0 | The WatchDog timer external signal enable bit<br>0: Disable<br>1: Enable |
| 4 | WdClock | 0x0 | The WatchDog timer clock source bit<br>0: PCLK<br>1: EXTCLK |

### 31.4.3.2 WdStatus (Offset = 0x10)

**Table 31-6. WdStatus Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 1 | WdStatus | R | The WdStatus register records if the WatchDog timer reaches 0 or not. It is read only. |

### 31.4.3.3 WdClear (Offset = 0x14)

**Table 31-7. WdClear Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 1 | WdClear | W | When writing 1 to this register, WdStatus will be cleared. |

### 31.4.3.4 WdIntrlen (Offset = 0x18)

**Table 31-8. WdIntrlen Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 8 | WdIntrlen | R/W | The WdIntrlen register decides the duration of the assertion of wd_rst, wd_intr, and wd_ext signals. The default value is 0xFF, which means that the default assertion duration of wd_rst, wd_intr, and wd_ext is 256 clock cycles. |

### 31.4.3.5 WdRevision (Offset = 0x1C)

**Table 31-9. WdRevision Register**

| Bit | Name | Type | Description |
|---|---|---|---|
| 32 | WdRevision | R | The revision number of FTWDT010 |

# Chapter 32

# Real Time Clock (RTC)

This chapter contains the following sections:


- 32.1 General Description
- 32.2 Features
- 32.3 Block Diagram
- 32.4 Block Description
- 32.5 Programming Model

## 32.1 General Description

Faraday Real Time Clock (FTRTC010) is a flexible, low-area, and low-power Real Time Clock (RTC), which accepts two clock sources: The APB bus clock (PCLK) and the EXTCLK (1-Hz clock frequency or any clock frequency). When the system enters the sleep mode, the PCLK clock can be gated by the system while the RTC keeps on counting. This mechanism promises the lowest power consumption when the system is asleep.

Furthermore, FTRTC010 provides the separate second, minute, hour, and day counters. The second counter is toggled each second, the minute counter is toggled each minute, the hour counter is toggled each hour, and the day counter is toggled each day. The second counter dominates most of the power consumption when the system is asleep. The second toggled bits have been separated to reduce the total power consumption when the system is in the sleep mode.  In addition, the separate counter mechanism reduces the complexity of the software in a way that the software does not have to calculate the minute, hour, or day information. It only needs to read the counter values and calculate the current time.

Faraday RTC provides a programmable auto-alarm function. When the second-auto-alarm function is turned on, the RTC will automatically trigger an interrupt each second. The automatic minute and hour alarm can also be turned on. This function is useful for implementing a clock.

## 32.2 Features

- PCLK gated by the system to save the power in the sleep mode.
- Separate second, minute, hour, and day counters that reduce the power consumption and the software complexity.
- Supports programmable automatic second, minute, and hour alarms

## 32.3 Block Diagram

Figure 32-1 shows the block diagram of Faraday RTC.

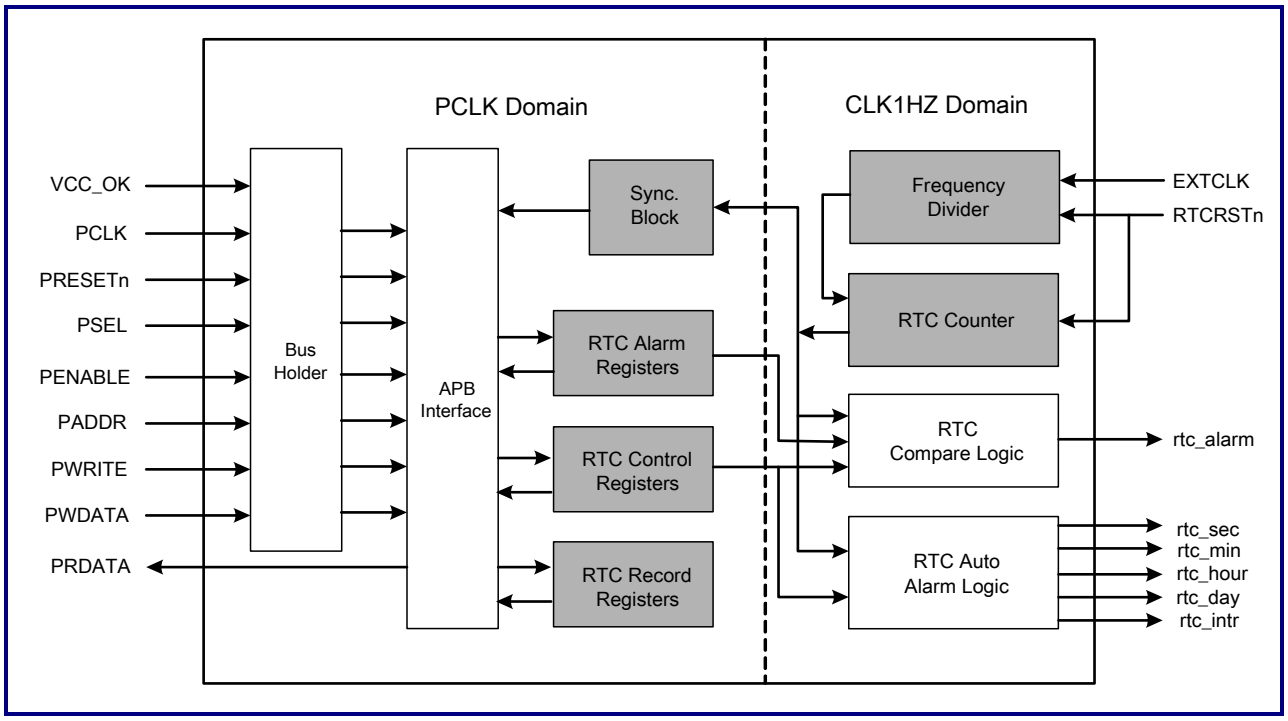



**Figure 32-1. Block Diagram**

## 32.4 Block Description

### 32.4.1 APB Interface

The APB interface accepts the standard APB bus signals. It meets the read/write requirements from the APB bus. When the PSEL signal is active, the APB interface decodes the PADDR signal to select one register. The PWRITE signal indicates a read or write operation. If the PWRITE signal indicates a write operation, PWDATA will be written to the selected register. If the PWRITE signal indicates a read operation, then the selected register will be read out and placed into PRDATA.

### 32.4.2 RTC Alarm Registers

The RTC alarm registers include the AlarmSecond, AlarmMinute, AlarmHour, and AlarmDays registers. The legal range of the AlarmSecond and AlarmMinute registers is from 0 to 59, while the legal range of the AlarmHour register is from 0 to 23. If either value of the AlarmSecond, AlarmMinute, or AlarmHour register exceeds its legal range, the rtc_alarm interrupt will not be triggered.

### 32.4.3 RTC Control Register

The RTC control register controls the RTC and the auto-alarm function. The default settings are that the RTC is disabled and the auto-alarm function is disabled.

### 32.4.4 RTC Record Register

The RTC record register only includes one RtcRecord register. When the system is initialized, users input the current time. The record value can be calculated by software by using the following expression:

***RtcDays*86400 + RtcHour*3600 + RtcMinute*60 + RtcSecond + RtcRecord***
***= seconds of (Current time - Base time)***
**** RtcHour, RtcMinute and RtcSecond are registers contained in RTC counter block.*
**** Base time is defined by programmer. For example, it can be defined at 2000/01/01/00:00:00.*
**** Current time is input by user when system is initialized*

After setting the RTC record register, the software can always use the above expression to calculate the current time.

### 32.4.5 Sync. Block

Since FTRTC010 has two clock domains that may propagate a metatable value when crossing the domain reading, the relationship between the PCLK and CLK1HZ clock domains can be expressed as below:

**PCLK / CLK1HZ > 4**

The following situations of the crossing reading need to be considered.

**Table 32-1. Crossing Clock Domain Signals**

| | |
|---|---|
| CLK1HZ domain → PCLK domain | RtcSecond, RtcMinute, RtcHour, and RtcDays registers:<br>The synchronization is made by the sync. block. |
| | rtc_alarm, rtc_sec, rtc_min, rtc_hour, and rtc_day interrupts:<br>The synchronization is made by the interrupt controller. |
| PCLK domain → CLK1HZ domain | Alarm registers and control register:<br>These registers are assumed to remain stable when the RTC is working. |

### 32.4.6 RTC Counters

The RTC counters include the second, minute, hour, and day counters. The second counter increased by second and rounded to zero when the value exceeds 59; the minute counter increased by minute and rounded to zero when the value exceeds 59; the hour counter increased by hour and rounded to zero when the value exceeds 23; the day counter increased by day. When RTCRSTn is reset, the counters are set to zero. These counters belong to the CLK1HZ clock domain and are read-only.

### 32.4.7 RTC Auto-alarm Logic

The RTC auto-alarm logic can automatically trigger an interrupt each second, each minute, each hour, or each day. For example, when the function of automatic second alarm is turned on, the RTC auto-alarm logic will trigger an interrupt by each second. This function is useful for implementing a RTC clock. The programmer can enable the auto-alarm function by writing a '1' to the RTC control register. Please refer to Section 3.2.9 for details about the RtcCR register.

### 32.4.8 RTC Compare Logic

The RTC compare logic is used to control the rtc_alarm interrupt. This logic block includes the second, minute, and hour comparators. If the AlarmSecond register equals to the value of the second counter, the AlarmMinute register equals to the value of the minute counter, and the AlarmHour register equals to the value of the hour counter, the rtc_alarm interrupt will then be triggered. The programmer can enable this logic block by writing a '1' to the RTC control register. Please refer to Section 3.2.9 for details about the RtcCR register.

### 32.4.9 Frequency Divider

Because the RTC counters belong to the 1-Hz clock domain, the EXTCLK clock domain, which is not a 1-Hz clock, must be divided into the 1 Hz clock domains by using the frequency divider. However, if EXTCLK is already a 1-Hz clock, users can disable the frequency divider, and the EXTCLK clock domain will be directly passed to the RTC counters.

## 32.5 Programming Model

### 32.5.1 Summary of Real Time Clock Registers

Table 32-2 lists the offset, type, width, reset value, name, and configuration option of each programmable RTC register.

**Table 32-2. RTC Register Summary**

| Offset | Type | Width | Reset | Name | Configuration | Description |
|---|---|---|---|---|---|---|
| 0x00 | R | 6 | 0x0 | RtcSecond | None | RTC second register |
| 0x04 | R | 6 | 0x0 | RtcMinute | None | RTC minute register |
| 0x08 | R | 5 | 0x0 | RtcHour | None | RTC hour register |
| 0x0C | R | 16 | 0x0 | RtcDays | None | RTC day count register |
| 0x10 | R/W | 6 | 0x3F | AlarmSecond | None | RTC second alarm register |
| 0x14 | R/W | 6 | 0x3F | AlarmMinute | None | RTC minute alarm register |
| 0x18 | R/W | 8 | 0x7F | AlarmHour | None | RTC hour alarm register |
| 0x20 | R/W | 7 | 0x0 | RtcCR | None | RTC control register |
| 0x24 | R/W | 6 | 0x0 | WRtcSecond | RTC_CONTWR | RTC second counter write port |
| 0x28 | R/W | 6 | 0x0 | WRtcMinute | RTC_CONTWR | RTC minute counter write port |
| 0x2C | R/W | 5 | 0x0 | WRtcHour | RTC_CONTWR | RTC hour counter write port |
| 0x30 | R/W | 16 | 0x0 | WRtcDays | RTC_CONTWR | RTC day counter write port |
| 0x34 | R/W | 5 | 0x0 | IntrState | None | RTC interrupt register |
| 0x38 | R/W | 32 | 0x0 | RtcDivide | None | RTC frequency divider |
| 0x3C | R | 32 | - | RtcRevision | None | RTC Revision Register |
| 0x40 | R | 32 | 0x0 | RtcCurrent | None | RTC Current time Register |

### 32.5.2 Register Descriptions

The following sections describe the RTC registers.

#### 32.5.2.1 RtcSecond

The RtcSecond register is the second counter register. When reset, the RtcSecond register is set to zero. Once the RTC is enabled, the value of the RtcSecond register is increased every second. When the value of the RtcSecond register exceeds 59, the value is reset to zero. The range of the RtcSecond register is from 0 to 59. If the RTC is disabled, the RtcSecond register will hold the value. The value of the RtcSecond register will be reloaded by the WRtcSecond register when the RtcCR[6] is set to '1.'

**Table 32-3. RtcSecond Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [5:0] | RtcSecond | Read | 0 | None |

#### 32.5.2.2 RtcMinute

The RtcMinute register is the minute counter register. When reset, the RtcMinute register is set to zero. Once the RTC is enabled, the value of the RtcMinute register is increased every minute. When the value of the RtcMinute register exceeds 59, the value is reset to zero. The range of the RtcMinute register is from 0 to 59. If the RTC is disabled, the RtcMinute register will hold the value. The value of the RtcMinute register will be reloaded by the WRtcMinute register when the RtcCR[6] is set to '1.'

**Table 32-4. RtcMinute Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [5:0] | RtcMinute | Read | 0 | None |

#### 32.5.2.3 RtcHour

The RtcHour register is the hour counter register. When reset, the RtcHour register is set to zero. Once the RTC is enabled, the value of the RtcHour register is increased every hour. When the value of the RtcHour register exceeds 23, the value is reset to zero. The range of the RtcHour register is from 0 to 23. If the RTC is disabled, the RtcHour register will hold the value. The value of the RtcHour register will be reloaded by the WRtcHour register when the RtcCR[6] is set to '1.'

**Table 32-5. RtcHour Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [4:0] | RtcHour | Read | 0 | None |

### 32.5.2.4 RtcDays

The RtcDays register is the day counter register. Before enabling the RTC, the programmer should input the current time and calculate the value of the RtcRecord register. After enabling the RTC, the value of the RtcDays register will always be increased by one day. This register will count the days from the time the RTC is enabled. The programmer can use this register to calculate the correct date when the system wakes up. The value of the RtcDays register will be reloaded by the WRtcDay register when the RtcCR[6] is set to '1.'

**Table 32-6. RtcDays Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [15:0] | RtcDays | Read | 0 | None |

### 32.5.2.5 AlarmSecond

The AlarmSecond is the second alarm register. If the programmer wants to trigger an rtc_alarm interrupt at 12:10:10 (hour:min:sec), the AlarmSecond register needs to be set to 0xA. If the value of the AlarmSecond register exceeds 0x3B, the rtc_alarm interrupt will not be triggered; however, the second counter will keep on counting.

**Table 32-7. AlarmSecond Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [5:0] | AlarmSecond | Read/Write | 0x3F | None |

### 32.5.2.6 AlarmMinute

The AlarmMinute register is the minute alarm register. If the programmer wants to trigger an rtc_alarm interrupt at 12:10:10 (hour:min:sec), the AlarmMinute register needs to be set to 0xA. If the value of the AlarmMinute register exceeds 0x3B, the rtc_alarm interrupt will not be triggered; however, the minute counter will keep on counting.

**Table 32-8. AlarmMinute Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [5:0] | AlarmMinute | Read/Write | 0x3F | None |

### 32.5.2.7 AlarmHour

The AlarmHour register is the hour alarm register. If the programmer wants to trigger an rtc_alarm interrupt at 12:10:10 (hour:min:sec), the AlarmHour register needs to be set to 0xC. If the value of the AlarmHour register exceeds 0x17, the rtc_alarm interrupt will not be triggered; however, the hour counter will keep on counting.

**Table 32-9. AlarmHour Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| 7 | PORpwm_period | Read/Write | 0x0 | None |
| [6:5] | POR_control | Read/Write | 0x3 | 0: Turn the POR off<br>1: Turn the POR into PWM mode<br>2: Reserved<br>3: Turn the POR on |
| [4:0] | AlarmHour | Read/Write | 0x1F | None |

### 32.5.2.8 RtcCR

The RtcCR register is the RTC control register which controls the RTC and the auto-alarm function. Bit 0 of the RtcCR register is the RTC enable bit. When disabled, the rtc_alarm interrupt will be gated to zero, and the RTC counters will be held. This means that the power consumption will approach zero when the RTC is disabled.

Bit 1 of the RtcCR register controls the second auto-alarm function. When turned on, the rtc_secarm interrupt will be triggered every second. Bit 2 of the RtcCR register controls the minute auto-alarm function. When turned on, the rtc_minarm interrupt will be triggered every minute. Bit 3 of the RtcCR register controls the hour auto-alarm function. When turned on, the rtc_hourarm interrupt will be triggered every hour. Bit 4 of the RtcCR register controls the day auto-alarm function. When turned on, the rtc_dayarm interrupt will be triggered every day. Bit 5 of the RtcCR register controls the RTC auto-alarm function. When turned on, the rtc_alarm interrupt will be triggered if all the RTC counters match the alarm registers.

Bit 6 of the RtcCR register controls the reload of the counter value. When turned on, the value of the WRtcSecond register will be loaded to the RtcSecond register, the value of the WRtcMinute register will be loaded to the RtcMinute register, the value of the WRtcHour register will be loaded to the RtcHour register, and the value of the WRtcDay register will be loaded to the RtcDays register. This bit will be cleared automatically after the values of the WRtcSecond, WRtcMinute, WRtcHour, and WRtcDay registers are loaded to the counters. Once the RtcCR[6] bit is set from high to low, the value of the RtcSecond register will be increased by 1. This bit is effective when the user defines the RTC_CONTWR macro.
The following table lists the bit assignment of the RtcCR register.

**Table 32-10. Control Register of RTC**

| Bit | Name | Description |
|---|---|---|
| 0 | RTC enable | RTC enable<br>0: Disable<br>1: Enable |
| 1 | RTC interrupt every second | RTC auto-alarm per second<br>When set to on, the rtc_sec interrupt is issued every second.<br>0: Disable<br>1: Enable |
| 2 | RTC interrupt every minute | RTC auto alarm per minute<br>When set to on, the rtc_min interrupt is issued every minute.<br>0: Disable<br>1: Enable |
| 3 | RTC interrupt every hour | RTC auto alarm per hour<br>When set to on, the rtc_hour interrupt is issued every hour.<br>0: Disable<br>1: Enable |
| 4 | RTC interrupt every day | RTC auto alarm per day<br>When set to on, the rtc_day interrupt is issued every hour.<br>0: Disable<br>1: Enable |
| 5 | RTC alarm interrupt | RTC alarm interrupt<br>When set to on, the rtc_alarm interrupt is issued if the RTC counters all match the RTC alarm registers. |

| Bit | Name | Description |
|---|---|---|
| 6 | RTC Counter Load | RTC counter load<br>When set to on, the counters can be reloaded.<br>0: Disable<br>1: Load |

### 32.5.2.9 WRtcSecond

The WRtcSecond register is the port to write the RtcSecond counter. When RtcCR[6] is set to '1,' the value of the WRtcSecond register will be loaded to the RtcSecond counter. This register is effective when the user defines the RTC_CONTWR macro.

**Table 32-11. WRtcSecond Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [5:0] | WRtcSecond | Read/Write | 0x0 | None |

### 32.5.2.10 WRtcMinute

The WRtcMinute register is the port to write the RtcMinute counter. When RtcCR[6] is set to '1,' the value of the WRtcMinute register will be loaded to the RtcMinute counter. This register is effective when the user defines the RTC_CONTWR macro.

**Table 32-12. WRtcMinute Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [5:0] | WRtcMinute | Read/Write | 0x0 | None |

### 32.5.2.11 WRtcHour

The WRtcHour register is the port to write the RtcHour counter. When RtcCR[6] is set to '1,' the value of the WRtcHour register will be loaded to the RtcHour counter. This register is effective when the user defines the RTC_CONTWR macro.

**Table 32-13. WRtcHour Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [4:0] | WRtcHour | Read/Write | 0x0 | None |

### 32.5.2.12 WRtcDay

The WRtcDay register is the port to write the RtcDays counter. When RtcCR[6] is set to '1,' the value of the WRtcDay register will be loaded to the RtcDays counter. This register is effective when the user defines the RTC_CONTWR macro.

**Table 32-14. WRtcDay Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [15:0] | WRtcDay | Read/Write | 0x0 | None |

### 32.5.2.13 IntrState

The IntrState register is the interrupt status register of RTC which records the status of each interrupt. Each bit of this register maps to one type of interrupt. This register must be cleared by the user.

**Table 32-15. IntrState Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| 0 | rtc_sec | Read/Write | 0x0 | This bit indicates that the rtc_sec interrupt has occurred. |
| 1 | rtc_min | Read/Write | 0x0 | This bit indicates that the rtc_min interrupt has occurred. |
| 2 | rtc_hour | Read/Write | 0x0 | This bit indicates that the rtc_hour interrupt has occurred. |
| 3 | rtc_day | Read/Write | 0x0 | This bit indicates that the rtc_day interrupt has occurred. |
| 4 | rtc_alarm | Read/Write | 0x0 | This bit indicates that the rtc_alarm interrupt has occurred. |

### 32.5.2.14 RtcDivide

The RtcDivide register represents the cycle number that the frequency divider can divide the EXTCLK clock into 1-Hz clock. If enable bit of the RtcDivide register is set to '1', the frequency divider will operate. Otherwise, the EXTCLK clock will be directly passed to the RTC counters. For example, if the value of the EXTCLK clock is 32.768 kHz, then DividerCycle of RtcDivide must be set to 0x8000 and DividerEnable must be set to '1'. The divider will divide the EXTCLK clock of 32.768 kHz into 1-Hz clocks. The default value of the RtcDivide register is 0x8000. If the EXTCLK clock is not 32.768 kHz, the RtcDivide register should be set to a correct value then the divider will divide the EXTCLK frequency clock into 1-Hz clock after 32768 EXTCLK cycles.

**Table 32-16. RtcDivide Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| 31 | DividerEnable | Read/Write | 0x0 | Enable bit<br>0: Disable Divider<br>1: Enable Divider |
| [30:0] | DividerCycle | Read/Write | 0x8000 | This field indicates the cycle number which the divider will divide EXTCLK into 1-Hz clocks. |

### 32.5.2.15 RtcRevision

The RtcRevision register records the version of RTC used by the user.

**Table 32-17. RtcRevision Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [31:0] | RtcRevision | Read | - | The version of RTC |

### 32.5.2.16 RtcCurrent

The RtcCurrent register is the current time register of RTC. The RtcCurrent register includes RtcSecond[5:0], RtcMinute[5:0], RtcHour[4:0], and RtcDay[14:0], in a total of 32 bits of information. The user can read the current time only with one APB read transaction.

**Table 32-18. RtcCurrent Register**

| Bit | Name | Type | Reset | Comment |
|---|---|---|---|---|
| [5:0] | RtcSecond[5:0] | Read | 0x0 | - |
| [11:6] | RtcMinute[5:0] | Read | 0x0 | - |
| [16:12] | RtcHour[4:0] | Read | 0x0 | - |
| [31:17] | RtcDay[14:0] | Read | 0x0 | - |

# Chapter 33

# DC Characteristics

This chapter contains the following sections:


- 33.1 Absolute Maximum Ratings
- 33.2 Recommended Operating Conditions
- 33.3 I/O Pad Capacitance
- 33.4 DC Characteristics of 3.3 V I/O Cells

## 33.1 Absolute Maximum Ratings

**Table 33-1. Absolute Maximum Ratings**

| Symbol | Parameter | Rating | Unit |
|---|---|---|---|
| VCC10 | 1.0 V core power supply | -0.3 ~ 1.1 | V |
| VCC18 | 1.8 V power supply | -0.3 ~ 1.98 | V |
| VCC25 | 2.5 V power supply of the I/O pin | -0.3 ~ 2.75 | V |
| VCC10A_DLL | 1.0 V analog power supply of the DDR2 DLL | -0.3 ~ 1.1 | V |
| VCC10A_PLL | 1.0 V power supply of PLL | -0.3 ~ 1.1 | V |
| VCC3IO_OSCH | 3.3 V power supply of the OSCH pin | -0.3 ~ 3.63 | V |
| VCC3IO | Input voltage of 3.3 V I/O cells | -0.3 ~ 3.63 | V |
| TSTG | Storage temperature | -40 ~ 150 | °C |

## 33.2 Recommended Operating Conditions

**Table 33-2. Recommended Operating Conditions**

| Symbol | Parameter | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| VCC10 | 1.0 V core power supply | 0.95 | 1.0 | 1.05 | V |
| VCC18 | 1.8 V power supply | 1.71 | 1.8 | 1.89 | V |
| VCC33 | 3.3 V power supply of the I/O pin | 3.13 | 3.3 | 3.46 | V |
| VCC3IO_OSCH | 3.3 V power supply of the OSCH pin | 3.13 | 3.3 | 3.46 | V |
| VCC10A_PLL | 1.0 V power supply of PLL | 0.95 | 1.0 | 1.05 | V |
| VCC3IO | Input voltage of 3.3 V I/O cells | 0 | 3.3 | 3.46 | V |
| $T_A$ | Ambient temperature | 0 | - | 75 | °C |

## 33.3 I/O Pad Capacitance

**Table 33-3. I/O Pad Capacitance**

| Symbol | Parameter | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $C_{IN}$ | Input capacitance | 3.3 V I/O | - | 2.0 | - | pF |
| $C_{OUT}$ | Output capacitance | - | - | 3.2 | - | pF |

## 33.4 DC Characteristics of 3.3 V I/O Cells

**Table 33-4. DC Characteristics of 3.3 V I/O Cells**

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{il}$ | Input low voltage | LVTTL | - | - | 0.8 | V |
| $V_{ih}$ | Input high voltage | LVTTL | 2.0 | - | - | V |
| $V_{t-}$ | Schmitt-trigger negative-to-threshold voltage | LVTTL | 0.8 | 1.1 | - | V |
| $V_{t+}$ | Schmitt-trigger positive-to-threshold voltage | LVTTL | - | 1.6 | 2.0 | V |
| $V_{ol}$ | Output low voltage | $I_{ol}$ = 2 mA ~ 16 mA | - | - | 0.4 | V |
| $V_{oh}$ | Output high voltage | $I_{oh}$ = -2 mA ~ -16 mA | 2.4 | - | - | V |
| $V_{opu}$ | Output pull-up voltage for I/O cells | | VCC3IO - 0.9 | - | - | V |
| $R_{pu}$ | Input pull-up resistance | $V_{in}$ = 0 V | 40 | 75 | 190 | kΩ |
| $R_{pd}$ | Input pull-down resistance | $V_{in}$ = 3.3 V | 40 | 75 | 190 | kΩ |
| $I_{in}$ | True 3.3 V I/O input leakage current | $V_{in}$ = 3.3 V or 0 V | - | ±1 | - | μA |
| | Input leakage current of 3.3-V with 5-V tolerance I/O cells | $V_{in}$ = 5 V or 0 V | - | ±1 | - | μA |
| $I_{OZ}$ | Tri-state output leakage current | - | -10 | ±1 | 10 | μA |

# Appendix A:Package Dimensions

## A.1. E-PAD TQFP-176 (20 x 20 x 1.0)

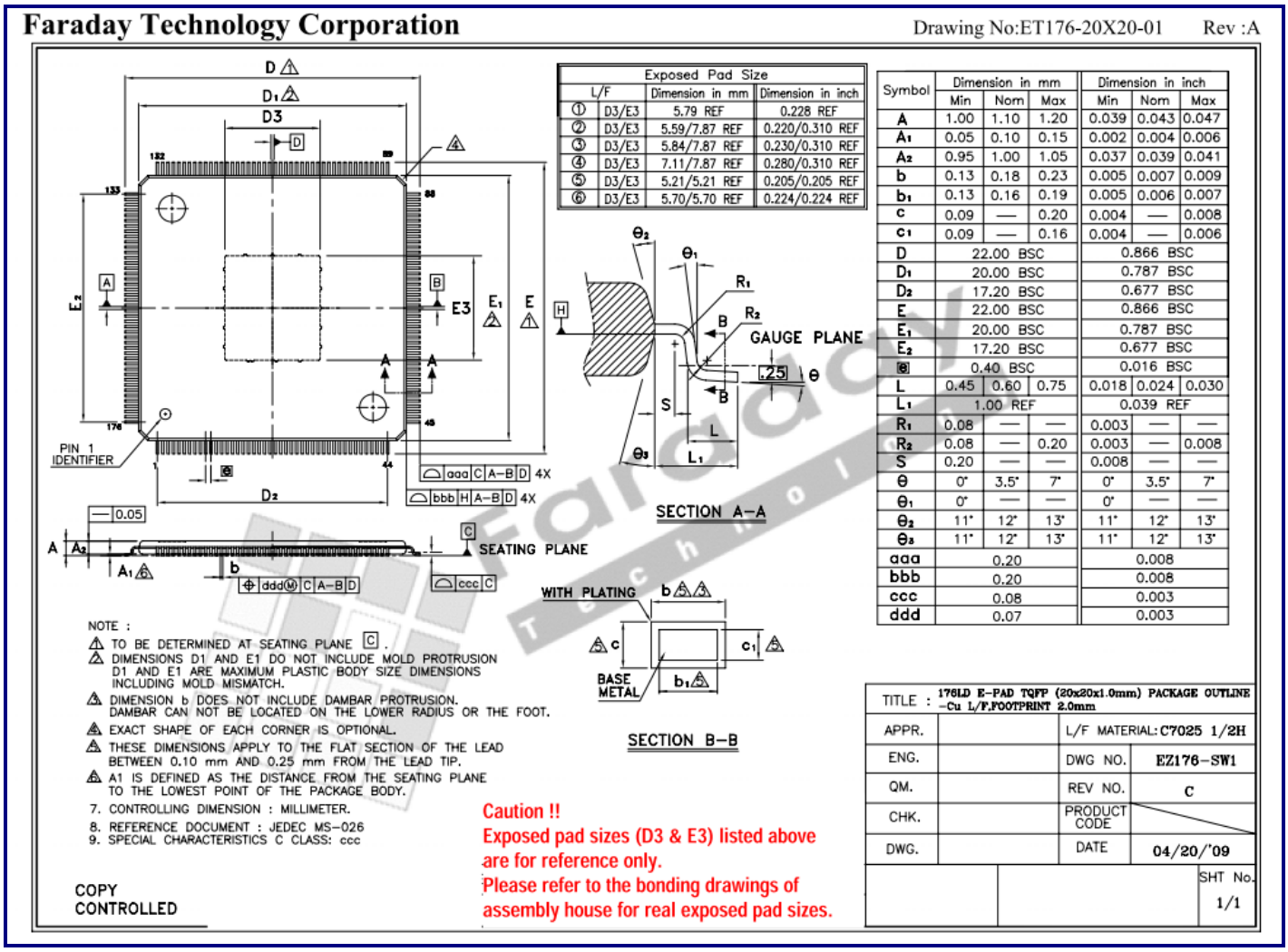
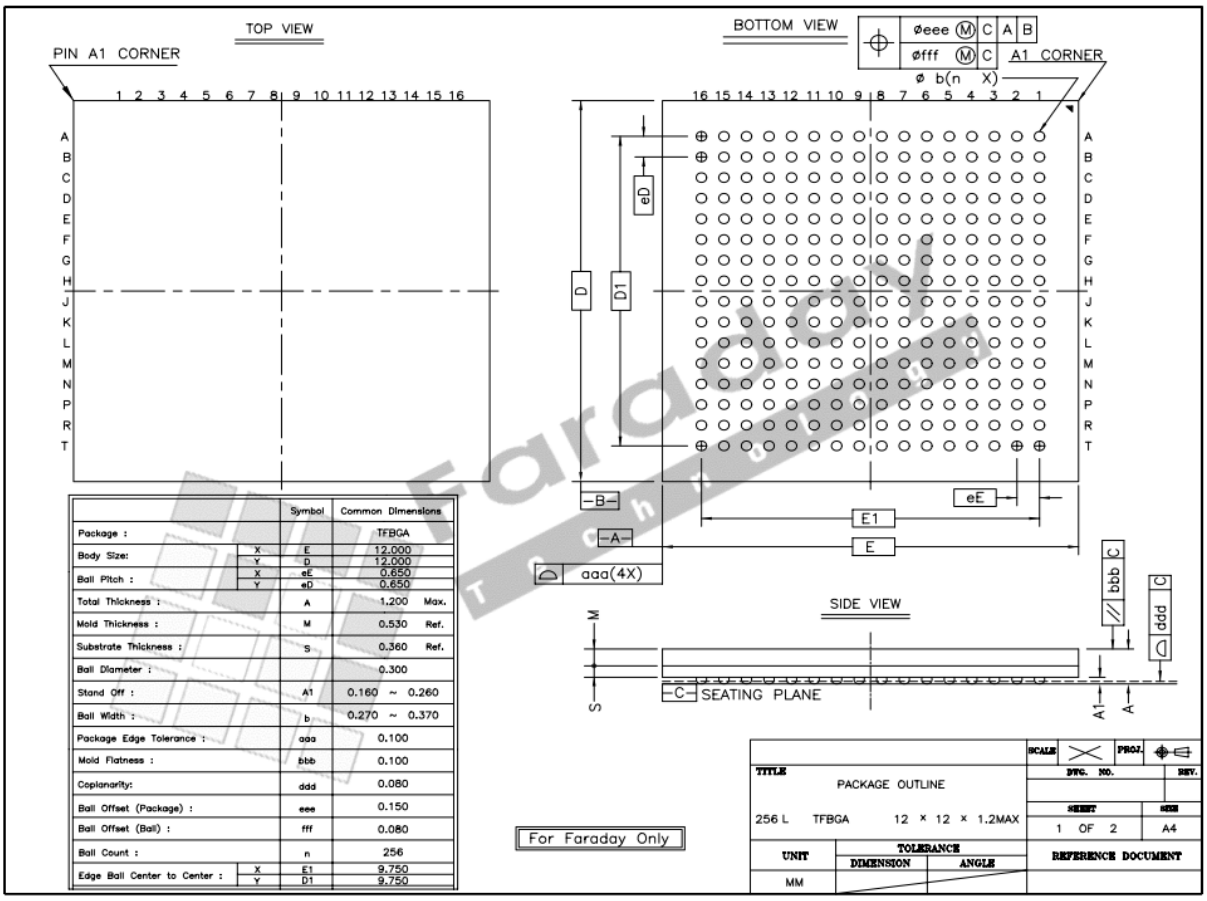



**Figure A-1. Package Outline of E-PAD TQFP-176**

## A.2. TFBGA-256 (12 x 12 x 1.2)




| | | Symbol | Common Dimensions |
|---|---|---|---|
| Package : | | | TFBGA |
| Body Size: | X | E | 12.000 |
| | Y | D | 12.000 |
| Ball Pitch : | X | eE | 0.650 |
| | Y | eD | 0.650 |
| Total Thickness : | | A | 1.200 Max. |
| Mold Thickness : | | M | 0.530 Ref. |
| Substrate Thickness : | | S | 0.360 Ref. |
| Ball Diameter : | | | 0.300 |
| Stand Off : | | A1 | 0.160 ~ 0.260 |
| Ball Width : | | b | 0.270 ~ 0.370 |
| Package Edge Tolerance : | | aaa | 0.100 |
| Mold Flatness : | | bbb | 0.100 |
| Coplanarity: | | ddd | 0.080 |
| Ball Offset (Package) : | | eee | 0.150 |
| Ball Offset (Ball) : | | fff | 0.080 |
| Ball Count : | | n | 256 |
| Edge Ball Center to Center : | X | E1 | 9.750 |
| | Y | D1 | 9.750 |

**Figure A-2. Package Outline of TFBGA-256**